University of Liège
Faculty of Applied Sciences
Department of Electrical Engineering and Computer Science

Characterization of neurodegenerative diseases with tree ensemble methods:

the case of Alzheimer's disease

Marie Wehenkel

Supervised by:
Pierre Geurts and Christophe Phillips

PhD dissertation

First printing, September 2018

The present dissertation has been evaluated by the members of the Jury (sorted by alphabetical order):

Dr. Christine Bastin		University of Liège, Belgium;
Prof. Danilo Bzdok		RWTH Aachen University, Germany;
Prof. Damien Ernst	(Chair)	University of Liège, Belgium;
Prof. Pierre Geurts	(Co-advisor)	University of Liège, Belgium;
Prof. Gilles Louppe		University of Liège, Belgium;
Dr. Christophe Phillips	(Advisor)	University of Liège, Belgium;
Prof. Yvan Saeys		Ghent University, Belgium.

The research described in the present dissertation was financially supported by the Belgian National Fund for Scientific Research (F.R.S.-FNRS).

The front-page figure is a modified version of the picture used for the following article: `http://cahooncare.com/3-stages-alzheimers-disease-cahoon-care-associates/`.

Characterization of neurodegenerative diseases with tree ensemble methods: the case of Alzheimer's disease

Wehenkel Marie

Abstract

For the last decade, the neuroscience field has observed the emergence of machine learning methods for the analysis of neuroimaging data. Unlike univariate methods that consider voxels one per one, these techniques analyse relationships between several voxels and are able to detect multivariate patterns. In the context of neurodegenerative diseases, such as Alzheimer's disease (AD), they can be used to design a diagnosis system and to find in neuroimages the patterns responsible for the disease. The context of the work presented here is thus the field of pattern recognition with neuroimaging. Our objective is to explore the possibilities that tree ensemble methods, such as Random Forests, offer in this domain in general, and in particular in the context of AD research. These methods suit very well the needs of this domain, as they combine very good predictive performances and provide interpretable results in the form of variable importance scores. Our contributions include both methodological developments around tree ensemble methods and applications of these methods on real datasets.

The methodological part of the thesis focuses on the analysis and the improvement of Random Forests variable importances for neuroimaging problems. Typical datasets in this domain are of very high dimensionality (hundreds of thousands of voxels) and contain comparatively very few samples (tens or hundreds of patients). Our first contribution is a theoretical and empirical analysis of how importance scores behave in such extreme settings, depending on the method parameters. We then propose several improvements of importance scores in such settings that take advantage of either the spatial structure between the features or a pre-defined partitioning of these features into groups. Finally, we address an issue with Random Forests importances, which is to find a threshold between truly relevant and irrelevant variables. For this purpose, we adapt several statistical methods proposed in the bioinformatics literature. These methods are extended to compute a statistical score for groups of features instead of individual features. This adaptation at the group level has been raised from our expectation to find groups of voxels explaining a disease instead of isolated voxels. We show that working at the group level leads to a higher statistical power than working at the feature level. The approach is applied on a real dataset for the prognosis of AD, where it is shown to highlight brain regions that are consistent with results in the literature.

In the second part of the thesis, we show different applications of Random Forests for AD research. First, we use tree-based ensemble methods in order to clinically characterize two different metabolic profiles observed in PET scans of AD patients. Second, we carry out an empirical comparison that shows that Random Forests are competitive with linear methods, in terms of accuracy and interpretability, on different real datasets related to three research questions about AD: the diagnosis of demented patients, the prognosis of mild cognitively impaired (MCI) patients, and the differentiation of MCI and AD patients.

Caractérisation de maladies neurodégénératives via des méthodes d'ensembles d'arbres: le cas de la maladie d'Alzheimer

Wehenkel Marie

Résumé

Depuis une dizaine d'années, le domaine des neurosciences a vu surgir l'analyse des données de neuroimagerie au moyen de méthodes d'apprentissage automatique. Contrairement aux méthodes univariées qui étudient les voxels séparément, ces nouvelles méthodes analysent les relations existant entre les variables et permettent ainsi la détection de motifs multivariés. Dans le cas des maladies neurodégénératives comme la maladie d'Alzheimer, l'utilisation de méthodes d'apprentissage peut donner lieu à des outils de diagnostic médical, qui analysent les images cérébrales et détectent les zones liées à la maladie. Le contexte de ce travail concerne donc le domaine de la détection de motifs dans des images cérébrales, via des méthodes d'ensembles d'arbres, telles que Random Forests, et ce, pour la maladie d'Alzheimer. Ces méthodes correspondent très bien aux besoins de ce domaine, en combinant de très bonne performance en prédiction and en fournissant des résultats interprétables via des scores d'importance. Nos contributions incluent à la fois des développements méthodologiques autour des méthodes d'ensembles d'arbres et des applications de ces méthodes sur des données réelles.

L'aspect méthodologique de cette dissertation concerne l'analyse et l'amélioration des scores d'importance fournis par les Random Forests dans le cas de problèmes de neuroimagerie. Les bases de données dans ce domaine sont en général à très haute dimension (des centaines de milliers de voxels) et contiennent comparativement très peu d'échantillons (des dizaines à quelques centaines de patients). Notre première contribution est donc une analyse théorique et empirique du comportement des scores d'importance, en fonction des paramètres de la méthode, dans de tels cas de figure. Nous proposons ensuite plusieurs améliorations des scores d'importance tenant compte de la structure spatiale entre les variables ou d'une partition a priori définie de ces variables en différents groupes. Finalement, nous adressons un problème rencontré avec l'utilisation des mesures d'importance qui concerne la détermination du seuil séparant les variables pertinentes des variables non pertinentes. Pour ce faire, nous adaptons plusieurs méthodes statistiques proposées dans la littérature en bioinformatique. En neuroimagerie, on s'attend à trouver des groupes de voxels expliquant une maladie plutôt que des voxels isolés. Ainsi, ces méthodes sont étendues de sorte à obtenir un score statistique pour des groupes de variables plutôt que pour des variables isolées. Nous montrons que travailler avec des groupes plutôt qu'avec des variables seules permet d'augmenter la puissance statistique des méthodes. Nous appliquons cette approche sur des données réelles pour le pronostic de la maladie d'Alzheimer. Les résultats soulignent des régions cérébrales cohérentes avec les résultats annoncés dans la littérature.

Dans la deuxième partie de cette thèse, nous travaillons sur différentes applications des Random Forests pour la recherche sur la maladie d'Alzheimer. D'une part, nous utilisons des approches d'ensembles d'arbres pour caractériser cliniquement deux différents profils métaboliques observés dans un en-

semble d'images PET de patients Alzheimer. D'autre part, nous montrons sur différents problèmes relatifs à la maladie d'Alzheimer la compétitivité des Random Forests en termes de performance et d'interprétabilité face aux méthodes linéaires. Nous étudions trois bases de données correspondant à différentes questions scientifiques sur la maladie d'Alzheimer: le diagnostic de patients déments, le pronostic des patients MCI, et la différenciation des patients MCI et Alzheimer.

Acknowledgements

A PhD thesis is not just a story of research but also a story of people. I would like to share here all my gratefulness to all the characters who participated in a certain way to the achievement of my personal journey.

First and foremost, I would like to thank my advisor and my co-advisor for their support along my whole PhD. In particular, I thank Christophe Phillips for sharing with me the importance of communicating, networking and travelling for research and I thank Pierre Geurts for his scientific guidance; he always pushed me to carry out the best research thanks to his precious advice and research creativity. I am so grateful to them.

I wish to express my gratitude to the members of the jury for their interest in my work. I sincerely appreciate the time they spent to read and evaluate this dissertation. I express a special thanks for Christine Bastin who participated to many discussions along this thesis. She provided me really valuable advice regarding the medical aspect of this work. This research project would have probably never seen the light without her.

I gratefully acknowledge the financial support from the Belgian National Fund for Scientific Research (FNRS) but also Rodolphe Sepulchre and Vincent Seutin for initiating me into the neuroscience field.

I heartily thank all my colleagues from the Montefiore Institute for providing me such a pleasant work place with all the smart (and sometimes less smart) discussions we had together. Thanks to the administrative staff, in particular to Sophie Cimino and Diane Zander for their constant help and kindness.

More specifically, thanks to the "machine learners" Antonio, Arnaud, Gilles, Jean-Mi, Laurine, Matthia, Raphaël, Rémy, Romain and Vân Anh. Thanks to Antonio for his unfailing availability and research enthusiasm. Thanks to Laurine for all the philosophical (or less philosophical) discussions at the second floor. Thanks to the "modellers" Raph, Julie, Alex, Guillaume and Bamdev. Thanks to my "office neighbours" Nicolas, Anthony and Anaïs. Thanks to my "more remote colleagues" Fred, Vincent, Aaron, David, Renaud and Delphine. Thanks to all of you (and those that I forgot) for having been there in the ups and downs of my PhD.

My deepest gratitude goes to my family and friends for their love and support. A warm thanks to my parents who always handled my nervous breakdowns, by being irreplaceable confidants and advisers in any crucial situation. Thanks to my brothers for their special support expressed sometimes with a joke about my research life or with a coffee/cookie break in my office.

My last words are for my love Florian, for his unfailing humour, support and love. It is priceless to have you by my side. You bring happiness and carefreeness in our life.

Contents

Part I

Machine learning for Alzheimer's disease

Chapter 1

Introduction

Begin at the beginning,the King said gravely, and go on till you come to the end: then stop.

- Lewis Carroll, *Alice in Wonderland*

1.1 Research context

Since the emergence of machine learning in the sixties, the number of applications using machine learning tools has never stopped to grow. The machine learning field is nowadays part of our lives. Indeed, it is involved in the recommendations we receive on a movie to watch, in the detection of credit-card fraud, in the recognition of different people on a picture, in the orientation of marketing campaigns. In the healthcare industry, machine learning methods provide automated diagnosis systems helping a physician in his diagnosis and his recommendations to a patient. To be efficient, such systems have to be developed in close collaboration with medical experts to validate the results and to have a deep understanding of the medical needs.

Alzheimer's disease is a complex brain disease still not completely understood. Indeed, still many research questions related to this disease stay unresolved, e.g. the ability to predict if an individual is likely to develop the disease or not several years later. In this field, machine learning can help to analyse brain images and to build interpretable diagnosis systems. One challenge in the application of machine learning methods is to cope with the very small size and the high dimension of the datasets commonly available in this domain. In this thesis, we study how a particular machine learning method family, tree-based ensemble methods, can contribute to answer research questions about Alzheimer's disease with brain imaging.

Given its interdisciplinary nature, this thesis work has been carried out at the University of Liège as a close collaboration between the Department of Electrical Engineering and Computer Science (Montefiore Institute) and the Cyclotron Research Centre. The Cyclotron Research Centre is a neuroscience research centre, where they notably perform brain imaging acquisitions, on human beings but also on animals, to study fundamental research questions related to, among others, memory and learning, circadian effects, ageing, and neurodegenerative diseases such as Alzheimer's disease. In this thesis, we take interest on Alzheimer's disease and benefit from several datasets provided by the Cyclotron Research Centre. Thanks to this collaboration, this four-year project was helped by the advice from physicians and neuro-psychologists in addition

to those of engineers.

We end this section by a brief description of Alzheimer's disease, the main research questions that surround it and a short motivation to use machine learning techniques in this domain. Section 1.2 then describes and motivates the main contributions of the thesis. We finally provide to the reader an outline of the manuscript in Section 1.3 with a brief description of the content of each chapter.

1.1.1 Alzheimer's disease

Alzheimer's disease (AD) is a *neurodegenerative* brain disease, i.e. neurons in some regions are irremediably destroyed during the course of the disease. The loss of these neurons leads to mental disorders and to memory dysfunctions.

This disease mainly affects elderly persons and has become increasingly frequent over the last decades as life expectancy in developed countries is continuously increasing [Brookmeyer et al., 2007]. The neurodegenerative disorder is currently not curable and, although some studies have focused on the development of medications delaying appearance of some symptoms, there does not exist at the moment anything to stop the disease progression.

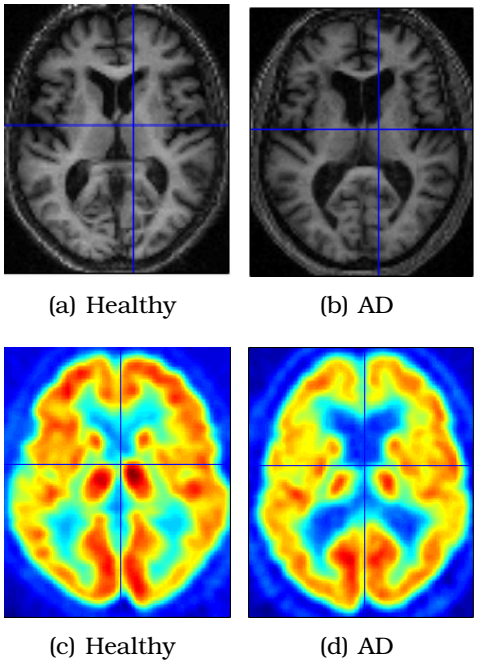
Patients are most commonly diagnosed with a clinical procedure designed for the measurement of cognitive impairments [Kelley and Petersen, 2007]. Disease assessment has been approved by a consortium (the American Psychiatric Association) which defined a precise list of criteria to meet in order to be declared as a diseased individual. These criteria concern notably the appearance of deficits in memory and troubles in executive functions (such as making plans, organizing activities,...). Clinical tests such as the *Mini-Mental State Examination* (MMSE) or the *Clinical Dementia Rating scale* (CDR) estimate precisely on a numeral scale the progression of cognitive impairment.

Once an individual is suspected of Alzheimer's disease, brain imaging is often used to figure out if disease progression has already caused brain damages. Indeed, consequences of the disease in the brain are clearly observable in MRI or PET data [Frisoni et al., 2010, Silverman et al., 2001]. MR images underline brain atrophy which is actually a normal consequence of ageing but is expressed much more severely in AD patients (cf. Figures 1.1(a) and 1.1(b)). In addition to structural information observed with structural MRI, FDG-PET scans bring information about functional deficit, which is illustrated in Figures 1.1(c) and 1.1(d).

Although the brain disease evolution is observable with medical imaging, people are most often diagnosed when brain damages are already substantial. An individual decides generally to have a consultation with a medical doctor when cognitive deficit begins to appear. However biological markers of the disease have started to progress up to several years earlier. Typical progression of AD biomarkers is shown in Figure 1.2, which shows that MRI (for brain structure) or PET (for $A\beta$ or Tau-neuronal injury and dysfunction)[1] images could be indicators of the disease well before the appearance of memory problems.

The *mild cognitive impairment (MCI)* clinical stage is a prodromal stage of the disease in which the patient begins to show some cognitive deficits but, according to clinical tests, the individual can not be already declared as demented. Each AD patient will go

[1] Amyloid PET imaging and FDG PET imaging are used for $A\beta$ deposit and energetic metabolism of the brain respectively.



(a) Healthy (b) AD

(c) Healthy (d) AD

Figure 1.1 – Structural MRIs of a healthy subject 1.1(a) and an AD patient 1.1(b) and FDG-PET scans of a healthy subject 1.1(c) and an AD patient 1.1(d). One can clearly see that the ventricles of the AD patient are enlarged and that the metabolic map of the AD patient presents some hypo-metabolic areas.

through a stage of mild cognitive impairment, whose duration is variable and unknown. However, not all MCI patients will necessarily develop Alzheimer's disease. Indeed, MCI patients are susceptible to develop other types of dementia (e.g. Lewy body dementia, Parkinson's dementia,...) or even to come back to the *cognitively normal* state.

The understanding of the MCI evolution towards the demented stage thus represents a big challenge in research. If the disease progression was predictable from a prodromal stage of the disease, treatments could be administrated sooner so as to delay the development of symptoms and so as to allow more clinical trials to be investigated to stop the disease progression. Moreover, family could be prepared sooner to the disease evolution. Machine learning could help to predict if the deficits of a MCI patient were likely or not to evolve into full blown AD. The sooner the disease is diagnosed, the sooner its progression and the development of symptoms could be stopped or delayed.

1.1.2 Machine learning in neuroimaging

The last few years have shown a growing interest for machine learning techniques in the neuroimaging field. The design of computer-aided diagnosis systems with machine learning (ML) algorithms have proved to be helpful not only to distinguish groups of subjects but also, as an alternative to univariate statistical tests, to provide interpretable information about the diagnosis, such as the brain areas affected by the disease. In neuroscience studies, the most popular ML approaches are based on linear models such as support vector machines [Hearst et al., 1998] with a linear kernel. Linear SVM models learned on the whole set of voxels from the images provide in general good accuracy and (limited) interpretability through voxel weight coefficients, e.g [Klöppel et al., 2008]. These coefficients can also be used for the construction of weight maps asso-

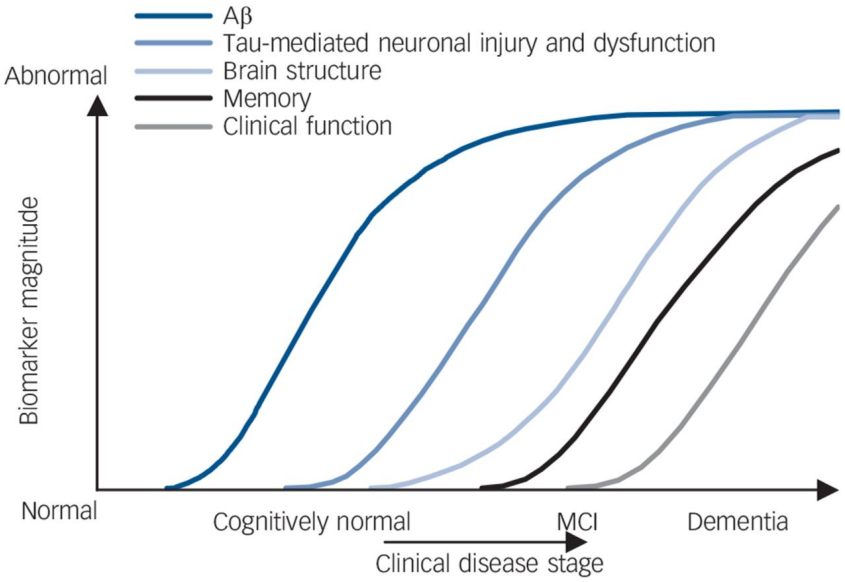


Figure 1.2 – Disease progression. Image taken from [Jack et al., 2010].

ciating a weight to each brain region and improving the interpretation of the diagnosis [Schrouff et al., 2013a].

More generally in machine learning, other well-known approaches are tree-based ensemble methods such as Random Forests [Breiman, 2001] or Extremely Randomized Trees [Geurts et al., 2006]. The idea of these methods is to average the predictions of several randomized decision trees in order to improve their accuracy by reducing their variance. Although these methods have not been studied extensively in the neuroimaging community, there are reports in the literature of their interesting behaviour [Langs et al., 2011, Richiardi et al., 2010]. The main advantages of these methods are their good predictive performance without (heavy) parameter tuning, even in very high dimensional settings such as in neuroimaging, and their interpretability through the derivation of variable importance and statistical scores from the forests [Huynh-Thu et al., 2012, Paul and Dupont, 2015].

1.2 Contributions of the dissertation

In the context exposed in the previous section, the objective of this thesis is to explore the opportunities that machine learning with tree ensemble methods offers in the field of neuroimaging in general and in particular in the context of fundamental research questions related to Alzheimer's disease. Our contributions are therefore of two different natures: first, methodological contributions around tree ensemble methods and their application in neuroimaging, second, more applicative contributions showing how these methods can be used to better understand Alzheimer's disease. We briefly motivate and describe here our main contributions, organized around these two research directions, and we link them to the different thesis chapters.

1.2.1 Tree ensemble methods in neuroimaging

In addition to predictive models, tree ensemble methods offer, through so-called variable importance scores, the possibility to highlight brain regions that are involved in some

conditions of interest. Our methodological developments focus mostly on the analysis and the improvement of these importance scores in the context of neuroimaging data.

One specificity of datasets in the neuroimaging domain is their very high dimensionality (hundreds of thousands of voxels) and their comparatively very low number of samples (tens or hundreds of patients). Our first contribution is a theoretical and empirical analysis of how importance scores behave in such an extreme setting (Chapter 4). Through a combinatorial analysis, we first derive theoretically the number of trees that are necessary to have observed each feature at least once during the forest construction (Section 4.2). This analysis thus gives the very minimum number of trees that should be built for a given problem not to miss any important feature. This lower bound is purely theoretical however and the number of trees required in practice to obtain reliable and stable importance scores is expected to be much higher. We thus complement the theoretical analysis with an empirical analysis, both on real and artificial datasets, of the impact of ensemble sizes on stability of importance scores, as measured through several metrics (Section 4.3). This analysis confirms the initial hypothesis that very large ensembles are necessary in practice to obtain stable importance scores on neuroimaging datasets.

Another specificity of neuroimaging datasets is the spatial neighbourhood structure that exists between the features (that indeed represent voxels in 3D images). Given this structure, one can reasonably assume that features that are spatially close should receive similar importance scores. As a consequence also of this structure, neuroscientists are more interested in highlighting brain regions, i.e., groups of contiguous voxels, than isolated voxels. These regions are usually predefined through anatomical atlases that segment the brain into regions of general interest. By exploiting such spatial structures or pre-defined groups, one can hope to reduce the required number of trees to obtain reliable and stable importance scores, with respect to what was predicted by the previous analyses. We make several contributions in this direction:

- First, we propose several alternative methods to derive better importance scores by taking into account either the spatial neighbourhood structure between the features or a pre-defined partitioning of these features into groups (Chapter 5). We divide these methods into preprocessing, embedded, or postprocessing methods depending on where the modification of the original algorithm is introduced. The benefit of these methods is illustrated on artificial and real datasets.
- As the most significant contribution of this thesis, we design a complete pipeline to select groups of features, each associated to the regions of a pre-defined atlas, by exploiting one of the alternative methods proposed in the previous study. This pipeline combines two ideas: (1) the aggregation of the importances of the individual group features to derive a group importance score and (2) the computation of an interpretable statistical score for each resulting group importance score through random permutation schemes. This latter step allows to more easily decide on a threshold to identify statistically significant groups. An extensive empirical analysis is performed to compare several instantiations of the aggregation operators and of the schemes to derive statistical scores. The potential of the approach both to identify the truly relevant groups and to improve predictive performance is shown through several experiments on artificial and real datasets (Chapters 6 and 7).

As a last methodological contribution, we carry out, on three real datasets, an empirical comparison of tree-based ensemble methods against linear methods commonly used in neuroimaging (Chapter 9). We consider also in this comparison variants of these methods that can exploit a prior grouping of the features. Methods are compared

in terms of predictive performance and in terms of their ability to identify the relevant features or groups.

1.2.2 Applications to Alzheimer's disease

In this dissertation, we take particular interest in exploiting our methodological developments to get a better understanding of Alzheimer's disease. Our main contributions towards this goal are the following:

- Throughout the thesis, we carry out our methodological developments and perform extensive experiments on a specific dataset provided by the Cyclotron Research Centre that contains FDG-PET images from stable MCI patients and from MCI patients that will eventually convert to Alzheimer's disease. When applied on this dataset, tree ensemble methods allow to build a predictive model for the prognosis of Alzheimer's disease and to identify brain regions that explain such prognosis. While this dataset is used in several chapters, the most extensive analysis of it is performed in Chapter 7.

- We contribute to a study aiming at characterizing, from a clinical point of view, two subtypes of Alzheimer's disease patients. The setting for this analysis is unusual. We dispose of two datasets: a dataset of FDG-PET scans of patients labelled according to their subtypes but without clinical information and a second dataset of FDG-PET scans of patients with clinical information but unlabelled in terms of subtypes. We propose a two-step approach to link clinical information to subtypes: a first tree ensemble model is trained on the first dataset and then used to predict subtypes in the second dataset. Variable importance scores are then computed on the second dataset to relate clinical features to the (predicted) subtypes. This analysis is described in Chapter 8.

- Our empirical comparison of ML methods on neuroimaging datasets in Chapter 9 is performed on three datasets related to three research questions about Alzheimer's disease: the diagnosis of demented patients, the prognosis of MCI patients, and the differentiation of MCI and AD patients. As a contribution to these questions, we report for each of them the best predictive performance that can be obtained with the considered methods and we analyse and compare the most relevant regions highlighted by these methods.

In the first two contributions, the analyses of the results have been carried out under the scrutiny of medical experts from the Cyclotron Research Centre and the medical relevance of the obtained results has been or will be analysed in greater depth by these experts in subsequent studies.

1.3 Outline, publications, and reading keys

The manuscript is divided into four parts.

Part I comprises this introduction and two chapters providing the necessary background. Chapter 2 introduces supervised learning and the different machine learning methods used in this thesis. Chapter 3 provides an introduction to neuroimaging modalities and methods. It also gives a brief overview of previous works in machine learning for neuroimaging and in the context of Alzheimer's disease. The datasets used

across the manuscript are also presented in this chapter.

Part II comprises four chapters focused on our methodological contributions. Chapter 4 contains the theoretical and empirical analysis of Random Forests importance scores in high dimensionality setting. Chapter 5 proposes and evaluates several techniques to improve the reliability of variable importance scores by taking into account spatial and group structure over the features.

Chapter 6 proposes a pipeline based on group selection and tree ensemble methods to design a computer aided prognosis system for Alzheimer's disease. This work is based on the following publications:

- M. Wehenkel, C. Bastin, P. Geurts, and C. Phillips. Computer Aided Diagnosis System Based on Random Forests for the Prognosis of Alzheimer's Disease. In *1st HBP Student Conference - Transdisciplinary Research Linking Neuroscience, Brain Medicine and Computer Science*, pages 14–18. Frontiers Media S.A., 2018a;
- M. Wehenkel, C. Bastin, C. Phillips, and P. Geurts. Tree ensemble methods and parcelling to identify brain areas related to Alzheimer's disease. In *Pattern Recognition in Neuroimaging (PRNI), 2017 International Workshop on*, pages 1–4. IEEE, 2017.

Chapter 7 explores further group importance scores and group selection methods with tree-based importance scores, also for the prognosis of Alzheimer's disease. This work is based on the following publication:

- M. Wehenkel, A. Sutera, C. Bastin, P. Geurts, and C. Phillips. Random Forests based group importance scores and their statistical interpretation: application for Alzheimer's disease. *Frontiers in Neuroscience*, 12:411, 2018b. doi: 10.3389/fnins.2018.00411.

Part III contains two more applicative chapters. Chapter 8 contains our work on the characterization of subtypes of Alzheimer's disease through clinical information, using a two-step approach. This work has been done in the context of a collaboration with the Cyclotron Research Centre. These results will be part of a publication in preparation for a journal:

- F. Meyer, M. Wehenkel, C. Phillips, P. Geurts, R. Hustinx, C. Bernard, C. Bastin, E. Salmon. *Characterization of a temporoparietal junction subtype of Alzheimer's disease.*

Chapter 9 compares several supervised learning methods on three different datasets related to three different questions around Alzheimer's disease.

The thesis is concluded by Chapter 10 with our main conclusions and perspectives.

There are also three appendices. Appendix A contains mathematical derivations related to Chapter 4. Appendix B reproduces supplementary results of Chapter 7. Appendix C gives details about the AAL atlas that is used in different chapters.

Chapter 2

Principles of machine learning

Chapter overview

In this chapter, we describe the machine learning background necessary to read comfortably through this thesis from chapter to chapter. We begin by explaining the concepts behind supervised learning, which is the subfield of machine learning we are concerned with. We explain how performances of machine learning algorithms can be assessed in general. Then we introduce tree-based ensemble methods, the machine learning techniques on which we focus in our work. As support vector machines are currently the most often used methods for pattern recognition in neuroimages, we thought important to explain principles behind them. Finally, we briefly describe other linear methods well adapted to our field of application.

2.1 Supervised learning

Machine learning is a sub-field of artificial intelligence focusing on the development and analysis of methods giving to a computer the ability to learn some difficult tasks from observed data. In particular, we take interest on *supervised learning* in this thesis. Supervised learning is a machine learning task in which a *model* is trained from observed data having been labelled for instance by a human. More formally, we are looking for a function f describing the relationship between the values of m inputs $(x_1, x_2, \ldots, x_m) = \mathbf{x} \in \mathcal{X}$, with $\mathcal{X}$ the input space, and the value of an output variable $\mathbf{y} \in \mathcal{Y}$, with $\mathcal{Y}$ the output space, such that:

$$f : \mathcal{X} \to \mathcal{Y}. \tag{2.1}$$

Such function f is subsequently exploited in order to predict the respective output for new input data.

The different instances of input and output data used to learn the function compose the *learning set* (or *training set*), denoted $LS = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{n}$. This set of input-output pairs gives rise to input and output matrices composed of as many rows as there are instances (also called *samples*) in the problem.

In this problem, m denotes the total number of features (variables), i.e., the size of the input vector, and n the total number of instances. If the output $\mathbf{y}$ contains more than one column, we talk about *multi-output problems*. In this thesis, we will only focus on single-output problems. In this case, the output $\mathbf{y}$ of a training instance is thus

a scalar value, and we will therefore drop the 'boldface vector notation' for the output variable. We therefore have typically a learning set which corresponds to $\{(\mathbf{x}_k, y_k)\}_{k=1}^n$ for a problem composed of n instances.

There exist two main classes of supervised learning problems: *classification* tasks and *regression* tasks. This depends on the output type. If y is qualitative (e.g. categorical or discrete output), we talk about classification whereas a quantitative output y corresponds to a regression problem. An example of regression task is, for instance, the estimation of the age of a person given its picture. Distinguishing cat pictures from other pictures represents a binary classification task while the classification of animals in cat, dog, rabbit and others is a multi-class problem.

In these examples, features can be the pixels of each image. Sometimes processing stages are also applied to the input features in order to broaden data exploitability. The task of finding suitable features describing a problem is called *feature engineering*.

2.2 Prediction error and model assessment

In order to assess the prediction performance of a model f, we need a quantitative measure of the discrepancy between a predicted label $f(\mathbf{x})$ and a true label y. Let us consider some loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ in order to characterize the prediction error involved by f. For a classification problem, the prediction error of f given a sample $(\mathbf{x}, y)$ is typically given by

$$L\left(f(\mathbf{x}), y\right) = I\left(y \neq f(\mathbf{x})\right), \tag{2.2}$$

where $I(.)$ is the indicator function (equal to 1 if its argument is true, 0 if it is not), while, for regression, a traditional choice is the squared-error loss

$$L\left(f(\mathbf{x}), y\right) = \left(y - f(\mathbf{x})\right)^2. \tag{2.3}$$

The aim of supervised learning is to find, from a learning sample LS, a function f of $\mathbf{x}$ that approximates at best the output y. By *best* function, we mean the function minimizing the *generalization error*, i.e. the expected value of the loss function

$$\mathbb{E}[L(f(\mathbf{x}), y)] \tag{2.4}$$

over instances $(\mathbf{x}, y)$ randomly drawn from the joint distribution $p(\mathbf{x}, y)$ over the input/output pairs.

However, the joint distribution of the input-output pairs is mostly unknown in real problems and thus one needs to estimate the generalization error from available data. It can be estimated from the training error, which is the average loss over the whole learning set LS

$$\overline{err} = \frac{1}{n} \sum_{i=1}^{n} L\left(f(\mathbf{x}_i), y_i\right). \tag{2.5}$$

As we will see in the subsequent sections, supervised learning algorithms can be viewed as optimization algorithms that search for a function f in some space of candidate functions (called the hypothesis space of the learning algorithm) so as to minimize the training error. More complex models correspond to larger hypothesis spaces, and lead typically to smaller training errors.

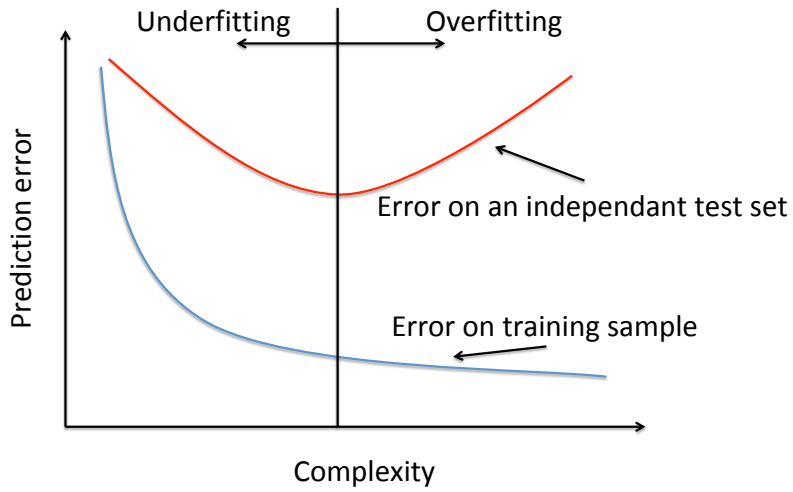


Figure 2.1 – Generalization and training error depending on the model complexity.

Therefore, training error is typically a rather bad estimator of the generalization error as it underestimates its value. Indeed, as illustrated in Figure 2.1, the training error will decrease with an increasing complexity of the model. If the model fits too much the training data, it may possibly fit also the noise or the randomness in the data. Consequently the model does not generalize well to new data and it is said to *overfit*. On the contrary, if the model is too simple, it will not capture enough the data characteristics. In this case, the model is said to *underfit*.

An alternative to estimate the generalization error is to create artificially a test sample from the original dataset. In a situation where a lot of samples are available, the best approach, called the *test set method*, is to randomly divide our dataset into a learning sample LS and a testing sample TS (e.g. proportion 70%/30% for learning/testing samples). The model is fitted on the learning sample and then its predictive performance is assessed by testing the model on the test set TS. When the initial dataset is very large, the performance estimated on TS is the same as if it was computed for a model learned on a dataset made up of both LS and TS.

However, such procedure would provide very bad estimation of the error with a small dataset. As it can be inferred from Figure 2.2, as the size of the training set is decreasing, the error is increasingly overestimated. Let us take an example of a dataset of one hundred samples. The accuracy is about 80%. A model trained on 70% of the data, i.e. on 70 samples only, will be significantly less accurate than a model learned on the whole dataset (around 75% instead of 80) and this effect is more and more remarkable with a decreasing size of the learning set.

Therefore, when the number of samples is low, the *cross-validation method* is recommended [Varoquaux et al., 2016]. The principle of the *K-fold* cross validation is the following. First, the whole dataset is divided into K parts of the same size. $K-1$ parts are used to learn the model and the model is estimated on the last part of samples that

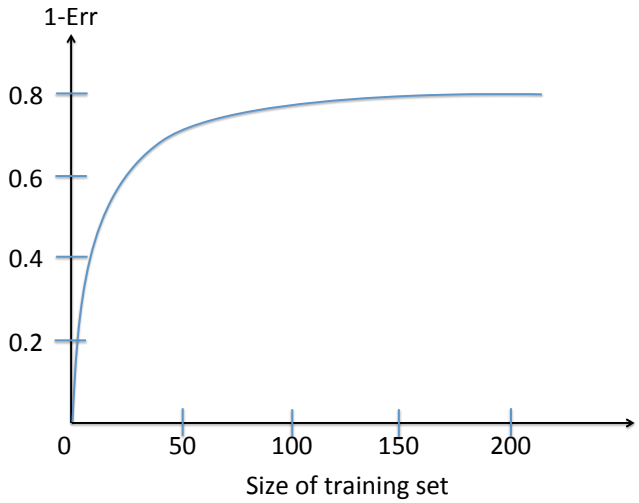


Figure 2.2 – Typical learning curve of a classifier learnt from a given dataset: evolution of the performance ($1 - Err$) depending on the size of the training set.

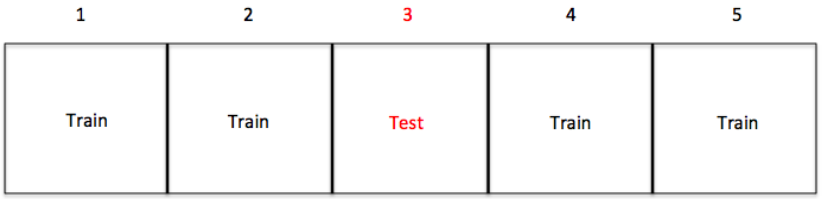


Figure 2.3 – Division of the learning sample in five folds for a five-fold cross validation.

has been put aside earlier. Then, the same procedure is repeated with another part aside and so on and so forth until every part has been used for testing a model. The error of a model trained on the whole set is thus estimated by the average error over the folds. Figure 2.3 shows an example of a cross validation procedure with 5 folds and, at this point, the third fold is the test set while the training set is composed of the other folds.

In the particular case $K = n$, we talk about a *leave-one-out* cross validation. A cross validation with $K = n$ will almost use the whole learning set to build its models but the different training sets will be nearly identical and this can lead to high variance of the performance estimates from one dataset to another. On the contrary, $K = 5$ or $K = 10$ has lower variance but is potentially biased (cf. learning curve in Figure 2.2). Moreover, to obtain an even more stable estimate of the model performance, the K-fold cross validation can be repeated many times (10 for instance) with different random separation of the dataset into K parts.

2.2.1 Performance metrics

Let us denote P the total number of samples belonging to the positive class and N the total number of samples belonging to the negative class, TP the total number of true positives detected by the method and TN the total number of true negatives detected by the method. The *accuracy* is the rate of samples that have been correctly classified, i.e.

$$Accuracy = \frac{TP + TN}{P + N}. \tag{2.6}$$

The *precision* is the proportion of positive samples correctly classified among all the samples that have been classified as positive. The *sensitivity* (also called true positive rate or recall) is the rate of positive samples that have been correctly classified while the *specificity* (or true negative rate) is the rate of negative samples that have been correctly classified. We thus have

$$Precision = \frac{TP}{TP + FP}, \tag{2.7}$$

$$Sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN}, \tag{2.8}$$

$$Specificity = \frac{TN}{N} = \frac{TN}{TN + FP}, \tag{2.9}$$

where FP denotes the number of false positives and FN denotes the number of false negatives.

Some methods provide in general the probability of a sample to belong to the positive class. For these methods, the different metrics depend on the threshold chosen to attribute the class of a sample. The *Receiver Operating Curve* (ROC) plots the evolution of the $sensitivity$ as a function of $1 - specificity$, i.e. the false positive rates, for varying thresholds. The *Precision-recall curve* (PR) plots the evolution of the precision as a function of the recall for varying thresholds. The area under these curves (AUC and AUPR respectively for the ROC and PR curves) can also be used as a performance metric, which does not require to choose a probability threshold.

2.3 Tree-based ensemble methods

We describe here one well-known family of machine learning methods: tree-based ensemble methods. With this section, we thus aim at laying the foundations of this thesis. Indeed, next chapters mainly focus on the behaviour of these methods for high dimensional data, like neuroimages.

2.3.1 Classification and regression trees

Before talking about tree ensembles, let us begin by single decision tree learning and the characteristics of a decision tree.

In 1984, Breiman et al. published their work about classification and regression trees (*CART*) [Breiman et al., 1984]. Although the output prediction is of different type for a classification or a regression problem, the principle is mainly the same for both cases.

A binary decision tree follows a tree-like structure in which each internal node is a binary test on a particular input variable and each leaf corresponds to a prediction of the output. The tests are also called the *splits* of the tree and each split gives thus rise

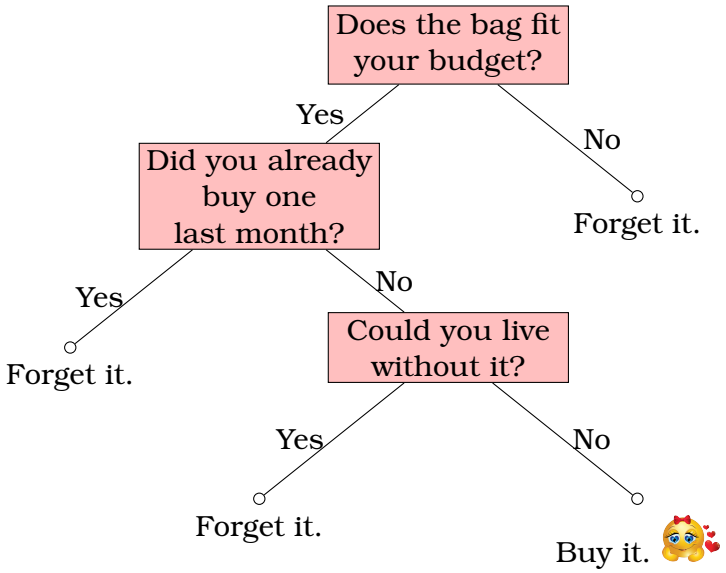


Figure 2.4 – Example of a decision tree: Should I buy this hand bag?

to two children only.

We illustrate the structure of a decision tree with a very simple example in Figure 2.4. In this example, the tree is composed of tree internal nodes with binary questions and leaves of the tree are the final decisions. This is a problem with only binary attributes and binary outputs. We notice that a decision tree model is easily understandable and we can easily propagate a new sample within the tree in order to obtain the output label.

Tree growing is a top-down procedure, i.e. it starts from the root node with the whole learning sample. At each node, an attribute and a value of the attribute is chosen to split the node and the number of samples in each branch decreases after each split.

Some criteria have to be chosen in order to decide at each node the best input variable to split. Moreover, for numerical input variables, binary splits consist of comparing the value of the variable to a threshold (called the cutpoint), so that the best cutpoint has also to be determined. More precisely, the algorithm of tree construction follows a *greedy* strategy. The choice of the input variable and cutpoint at a node is obtained by optimizing such criterion at this node and thus the solution is locally optimal at each stage. The algorithm considers at each step all the input variables and cutpoints possible. For numerical attributes, values are actually discretized in a fixed number of possibilities depending on the learning sample.

In particular, at a node $\mathcal{N}$, the split chosen will be the one that maximizes the expected reduction of impurity

$$\Delta I(\mathcal{N}) = I(\mathcal{N}) - \frac{n_l}{n_\mathcal{N}} I(\mathcal{N}_l) - \frac{n_r}{n_\mathcal{N}} I(\mathcal{N}_r), \tag{2.10}$$

where $I(\cdot)$ is the impurity function measuring the uncertainty about the output in a subsample, $n_\mathcal{N}$ the total number of examples at node $\mathcal{N}$ and n_l and n_r are the number of examples assigned by the split respectively to the left and right children $\mathcal{N}_l$ and $\mathcal{N}_r$ of node $\mathcal{N}$.

Tree growing is stopped when impurity cannot be decreased any more; we say that such a tree is *fully grown*. More precisely, a node is not developed any further if it is only composed of samples with the same output value (the node is then entitled *pure*) or if all attributes are constant in this leaf. The final output prediction of the leaves is the average value of the output label in the regression case while, for a classification problem, it is a majority vote.

For a regression problem, the impurity function at node $\mathcal{N}$ is commonly the variance of the output y for the sub-sample considered at node $\mathcal{N}$:

$$I(\mathcal{N}) = \frac{1}{n_{\mathcal{N}}} \sum_{i \in \mathcal{N}} (y_i - \overline{y})^2, \tag{2.11}$$

where $\overline{y} = \frac{1}{n_{\mathcal{N}}} \sum_{i \in \mathcal{N}} y_i$ is the average value of y for the instances $i \in \mathcal{N}$, i.e. the instances reaching the node $\mathcal{N}$.

In classification, two measures are mainly advised: the Gini index and the Shannon entropy. For a problem with C classes, the Gini index at node $\mathcal{N}$ is such that

$$I(\mathcal{N}) = 1 - \sum_{i=1}^{C} (p(c_i))^2, \tag{2.12}$$

where $p(c_i)$ denotes the frequency of occurrence of class c_i among the examples at $\mathcal{N}$, while the Shannon entropy at node $\mathcal{N}$ is defined by

$$I(\mathcal{N}) = -\sum_{i=1}^{C} p(c_i) \log_2 \left(p(c_i)\right). \tag{2.13}$$

In general, fully grown trees overfit the data because they often encode noisy information. Indeed, with a number of tests equal to $(\#samples - 1)$, it is possible to categorize perfectly every instance. However, some so-called pruning strategies can help to optimally reduce the size of the tree. If they are applied during the process of learning, we talk about *pre-pruning* whereas size reduction after learning is called *post-pruning*.

For pre-pruning a tree, one possibility is to stop the construction when the impurity reduction is inferior to some threshold. More traditionally, tree growing stops as soon the nodes contain a sufficiently small number of samples.

The post-pruning procedure consists in growing a fully grown tree on a portion of the learning set and using the rest of the learning set to evaluate the performance of this tree and all the sub-trees that it is made up of. The final selected tree is thus the pruned tree of minimal error on remaining samples.

2.3.2 Tree based ensemble methods

Ensemble methods are a type of machine learning technique that consists in combining the predictions of several models in order to improve the performance with respect to the use of a single one. In the context of tree based models, the idea is to fit many different trees and then to compute the final prediction for a new sample by combining in some suitable way the predictions obtained by propagating this sample in each individual tree.

We describe below the most popular tree-based ensemble methods. Actually, some of these methods could also be applied to any other machine learning algorithm but here we explain them in our context of tree based models.

All ensemble methods have a common parameter, which is the number T of trees used to compose the ensemble. Most of the time, this parameter is fixed to hundreds or even thousands of trees.

Tree Bagging Bagging, for *bootstrap aggregating*, has been proposed by Leo Breiman in 1996 [Breiman, 1996]. He proposed in his work to perturb input data by drawing *bootstrap samples*. A bootstrap sample of the learning set LS of size n is generated by randomly drawing with replacement n samples of the original data in LS. A model composed of T trees is thus obtained by growing T single (decision or regression) trees, each of them on a new bootstrap sample of the training data. The predictions of these trees are combined by using either majority vote (in the case of classification problems) or by simple averaging (in the case of regression problems).

The main idea defended by this procedure is the reduction of the high learning variance of single tree models. One can indeed show that the variance of tree bagging decreases monotonically when T increases, while its bias remains constant. This property justifies to use as high as possible values of T as permitted by the available computing budget. In order to yield at the same time a small bias, the individual trees are generally unpruned.

Random Forests In 2001, Leo Breiman published a new machine learning algorithm, called *Random Forests*, based on a combination of bagging and feature subsampling [Breiman, 2001]. In models generated by this algorithm, the individual trees of the ensemble are also learnt on different bootstrap copies of the training data, as for Bagging. An additional randomization is however introduced inside the tree growing procedure. Namely, the best split is no longer evaluated by seeking the best feature among all the input variables but it is chosen among a random subset of K input variables only. Before the construction of each split, a new subset of K variables is thus drawn. Such procedure further enforces diversity in the choice of splitting variables.

K is a parameter of the method and, so its value influences the precision of a model. For $K = m$, the method reduces to tree bagging. If K tends towards 1, variance reduces but the bias increases as the model is less and less fitted to the data themselves. Another characteristic is the improvement of the computational speed in learning with a small K value. We thus face a bias/variance trade-off and, in practice, it has been shown notably in [Geurts et al., 2006] that choosing K as the square root of the total number m of variables in the problem provides most often accurate and satisfying classifiers. This value is therefore often used as the default one.

Extremely randomized trees Geurts et al. [2006] suggested another approach to build models based on forests of trees. Unlike Random Forests, it is not based on bagging and uses the whole learning set for growing each tree. However, similarly to Random Forests, each splitting variable is the best one among K features randomly drawn.

In addition, one more added randomization characteristic of this method is in the choice of the cutpoint values used to split a node. For each of the K possible attributes, only one randomly chosen cutpoint is evaluated. This reduces the number of splits to consider and therefore computing times. The resulting method is called *Extremely randomized trees* (also named *Extra-trees*).

In summary, the above tree ensemble methods use random subsets of the samples and/or the variables in order to make the model more robust to changes in the dataset. In consequence, these methods exhibit a lower variance than a single decision tree: the models they provide therefore generalize better and overfit less the data. The perturbation introduced in the learning sample is however responsible for a slight increase of the bias most of the time.

Tree Boosting *Boosting* is another ensemble method that, unlike the other ones presented above, fits the models sequentially and not independently, on modified versions of the learning sample. Predictions are then combined by a weighted sum and not a simple average. Moreover, as the main idea is to combine many weak classifiers to make a good one, the ensemble is not composed of fully grown trees but of small trees of limited depth (e.g., *stumps*, trees composed of a single split). The most famous boosting algorithm is *AdaBoost* introduced by Freund and Schapire [1995] and consists in modifying at each iteration weights attributed to each sample depending on the prediction error associated to this sample. Weights are increased for misclassified samples and they are then taken into account in the learning process when the samples are counted for the computation of impurity decrease.

2.3.3 Interpretability and the importances of input variables

By its structure itself, a single tree model is actually easily interpretable. It makes sense that the tests close to the top of the tree influence a lot the final prediction while tests of lower level depend generally on variables of lower importance.

Unfortunately, such intuition is totally lost for a forest of trees as there is much diversity among trees. Moreover, forests are often composed of hundreds to thousands of trees and the analysis of each tree individually is completely intractable. For this reason, we cannot infer directly interpretations from the model observation. However, several methods have been proposed to derive variable importance scores from tree ensembles.

One of them is called the mean decrease of impurity (MDI). It consists in evaluating splits in a decision tree by the decrease of impurity resulting from the test. This quantity for one particular splitting variable is then accumulated for each split (weighted by the nodewise sub-sample size) in which the variable is used over the whole forest. This sum actually reflects the importance of the variable in the final prediction. The operation can be repeated for each input variable of the problem to provide an *importance score* for each one.

Mathematically, we denote by $\mathcal{I}(x_i, \mathcal{T}_j)$ the importance of a variable x_i $(\forall i = 1, \ldots, m)$ in a single tree $\mathcal{T}_j$ and this quantity is given by the mean decrease of impurity measure such that

$$\mathcal{I}(x_i, \mathcal{T}_j) = \sum_{\mathcal{N} \in \mathcal{T}_j | v(\mathcal{N}) = x_i} \frac{n_{\mathcal{N}}}{n_{\mathcal{T}_j}} \Delta I(\mathcal{N}), \tag{2.14}$$

where $v(\mathcal{N})$ denotes the variable used to split the node $\mathcal{N}$, $n_{\mathcal{T}_j}$ the number of samples in the learning set, $n_{\mathcal{N}}$ the number of samples reaching $\mathcal{N}$ and $\Delta I(\mathcal{N})$ has been defined earlier in Equation (2.10).

For a forest of T trees, the mean decrease of impurity measure of a variable x_i is simply averaged over all the trees in the forest. The importance score of the feature x_i is thus given by

$$\mathcal{I}(x_i) = \frac{1}{T}\sum_{j=1}^{T}\mathcal{I}(x_i, \mathcal{T}_j). \tag{2.15}$$

Intuitively, a feature will get a high importance score if it appears frequently in the forest and at top nodes (leading to large $\frac{n(\mathcal{N})}{n}$ ratios) and if it strongly reduces impurity at the nodes where it appears.

We provide in Algorithm 1 a pseudo-code for the building of a forest of T trees, with the standard Random Forests algorithm described earlier, and the generation of the importance scores for every feature in the learning sample LS.

Algorithm 1 Random Forests algorithm and feature importance scores generation.

Require: A learning sample LS (of size n), the number of selected features K, and a forest size T.

$\mathcal{I}(x_i) = 0, \forall i = 1, \ldots, m.$ ▷ Considered as global variables
for $t = 1$ to T **do**
    Generate a bootstrap sample LS^b from LS.
    Learn_a_randomized_tree(LS^b)
end for
$\mathcal{I}(x_i) \leftarrow \frac{1}{Tn}\mathcal{I}(x_i), \forall i = 1, \ldots, m.$

function LEARN_A_RANDOMIZED_TREE(LS)
    if all objects from LS have the same class **then**
        Create a leaf with that class.
    else
        Randomly pick K features.
        Evaluate the expected reduction of impurity $\Delta I(\mathcal{N})$ provided by the best split on each feature x_i among K at this node $\mathcal{N}$.
        Select the feature x_i^* giving rise to the maximum $\Delta I(\mathcal{N})$.
        $\mathcal{I}(x_i^*) \leftarrow \mathcal{I}(x_i^*) + n_{\mathcal{N}}\Delta I(\mathcal{N})$.
        Create a test node for the selected split and divide LS into sub-samples LS_1 and LS_2 according to this split.
        LEARN_A_RANDOMIZED_TREE(LS_1)
        LEARN_A_RANDOMIZED_TREE(LS_2)
    end if
end function

It is worth mentioning that, for Bagging and Random Forests, Breiman [2001] proposed an alternative measure that computes for each feature the mean decrease of accuracy (MDA) of the forest when the values of this feature are randomly permuted in the *out-of-bag* samples. In bagging, only a subset of the original sample is used for fitting each tree. The out-of-bag sample refers to the instances not used during the learning process and there is one out-of-bag sample for each tree of the forest. The concept is thus to use these samples to compute the prediction error involved by the forest model. An importance score for a variable x_i is thus associated by computing the prediction error when this variable is permuted and looking at the difference between the error before and after permutation. Both MDI and MDA measures are used in practice. Experimental studies [Strobl et al., 2007] have shown that the MDI is biased towards

features with a large number of values but this bias is irrelevant in our neuroimaging setting in this thesis, where all features are numerical. The MDI measure furthermore benefits from interesting theoretical properties in asymptotic conditions [Louppe et al., 2013] and is usually faster to compute as it does not require to perform random permutations.

Because of their ease of use, their robustness with respect to parameter tuning, their performance and their interpretability, tree-based ensemble methods are widely used in practice, notably for biomedical problems necessitating interpretation of results as well as accuracy. In bioinformatics, for instance, Random Forests are often used for biomarker discovery or genome-wide association studies [Díaz-Uriarte and De Andres, 2006, Lunetta et al., 2004, Genuer et al., 2010, Botta et al., 2014].

Variable importance scores can be useful for feature selection approaches to identify the (most) relevant variables related to a problem.

Definition 1. *According to [Guyon and Elisseeff, 2006], a variable x_i is said irrelevant with respect to the output y if for all subsets of features $B \subseteq V \setminus \{x_i\}$,*

$$P(x_i, y \mid B) = P(x_i \mid B)P(y \mid B),$$

where V is the set of input variables. A variable is relevant if it is not irrelevant.

In words, a variable x_i is relevant with respect to the output if there is a least one subset of variables B such that the output y depends on x_i conditioned on B. Relevant variable are thus variables that bring some information about the output in at least one context represented by the conditioning. On the other hand, irrelevant variables never explain the output in any conditioning. One common problem of feature selection consists in identifying all the relevant features [Nilsson et al., 2007, Kursa and Rudnicki, 2011, Sutera et al., 2018]. In [Louppe et al., 2013, Sutera et al., 2018], the authors have linked through several theorems importance scores with the notion of variable relevance, in asymptotic setting (ie., an infinite number of trees and samples). In this setting, they have for example shown that a variable is irrelevant if and only if its importance score as computed from a forests of totally randomized trees (i.e., grown with $K = 1$) is zero. When $K > 1$, a zero importance remains a necessary condition for a variable to be irrelevant but it is not a sufficient condition anymore as relevant variables can receive zero importances. In finite setting however, irrelevant variables can receive positive importance scores and one has thus to resort to statistical tests to distinguish the limit between truly relevant and irrelevant variables in variable importance rankings [Huynh-Thu et al., 2012, Genuer et al., 2010].

In many applications like in neuroimaging, variables can be highly correlated among themselves. These variables are said to be *partially redundant* when they share partially similar information about the output variable. Two variables are *totally redundant* if they are perfectly correlated. Chapter 7 of [Louppe, 2014] showed that redundancy can have a considerable effect on importance measures. If a copy x'_i of a variable x_i is added to the feature set, the importance score of x_i will decrease. Moreover, it will also impact the importance values of the other variables. It is therefore important to keep in mind such effects when we deal with correlated variables, as it is expected to be the case with neuroimaging applications.

2.4 Linear SVMs and other linear methods

In biomedical applications and in particular in the neuroimaging field, a popular method is *Support Vector Machines* [Cortes and Vapnik, 1995, Hearst et al., 1998] in its linear

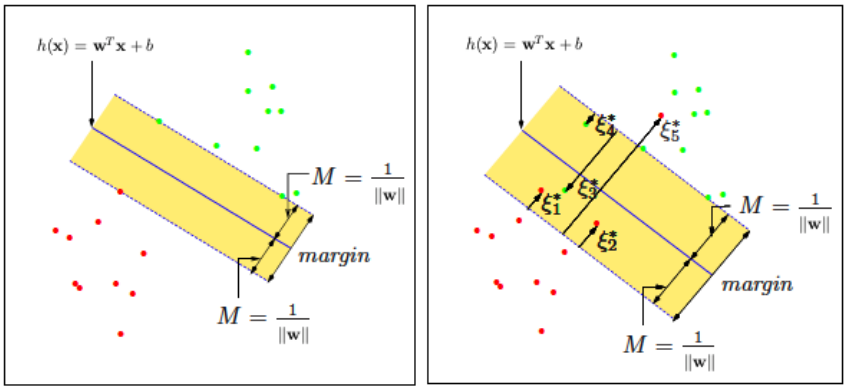


Figure 2.5 – Support Vector Machines. Left panel represents the hard-margin problem, in which samples are perfectly separable whereas the right panel illustrates the soft-margin problem, in which slack variables have to be introduced. Figure adapted from Hastie et al. [2009].

form. We thus introduce in the present section this method together with some other interesting linear methods.

The main concept of linear SVM is to find the hyperplane that best separates the data in different classes through the resolution of an optimization problem. As we focus on classification problems in this thesis, we only deal with the development for classification in this part of the manuscript.

2.4.1 Linear Support Vector Machines for binary classification

Let us consider the learning set $LS = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, n\}$ where $y_i \in \{-1, 1\}$, i.e. a two-class problem.

We want to find a decision function

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b) \tag{2.16}$$

minimizing the classification error on the learning set $\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq f(\mathbf{x}))$.

If the classes are perfectly separable, it is possible to find such a function fulfilling at the same time the condition

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0 \tag{2.17}$$

for all i from 1 to n.

Moreover, the optimal hyperplane $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ separating the two classes is defined in SVM as the one that maximizes the distance between this hyperplane and the nearest point of any class in LS. This distance is called the *margin* M and such hyperplane is called the *maximum margin hyperplane* and those nearest points called *support vectors*. The *hard-margin* problem is illustrated in the left panel of Figure 2.5.

The decision boundary function can thus be obtained by solving the following optimization problem

$$\max_{\mathbf{w},b,\|\mathbf{w}\|=1} M \tag{2.18}$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq M, i = 1, \ldots, n \tag{2.19}$$

which looks for the highest margin M with all the dataset points at least localized at a distance M from the hyperplane.

The condition $\| \mathbf{w} \| = 1$ can be removed by injecting it in (2.19) which thus becomes

$$\frac{1}{\| \mathbf{w} \|} y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq M. \tag{2.20}$$

Finally, we can arbitrarily fix $\| \mathbf{w} \| = \frac{1}{M}$ because, if $\mathbf{w}$ and b fulfil these conditions, their rescaling will not modify the value of the margin M.

Hence, the optimization problem to solve can be formulated as

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{2.21}$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, \ldots, n. \tag{2.22}$$

The dual formulation is obtained by using the Lagrangian

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] \tag{2.23}$$

with $\alpha_i \geq 0 \ \forall i$. It has to be minimized with respect to $\mathbf{w}$ and b and maximized with respect to α.

Optimal conditions can thus be obtained by deriving the Lagrangian with respect to $\mathbf{w}$ and b and setting these derivatives to zero. In particular, these operations provide the following conditions

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \tag{2.24}$$

$$0 = \sum_{i=1}^{n} \alpha_i y_i, \tag{2.25}$$

with $\alpha_i \geq 0$. These results can be substituted in the Lagrangian to give the following dual optimization problem

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \mathbf{x}_i^T \mathbf{x}_k \tag{2.26}$$

$$\text{subject to } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{2.27}$$

According to the *Karush-Tucker* conditions, the solution must also fulfil the condition

$$\alpha_i [y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1] = 0 \ \forall i, \tag{2.28}$$

which gives $\alpha_i = 0$ if $y_i(\mathbf{w}^T\mathbf{x}_i + b) > 1$ and x_i is not on the boundary of the margin. On the contrary, if $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$, $\alpha_i > 0$ and x_i is on the boundary. In this case, x_i is called a *support vector*. Any of these vectors can be used to obtain the value of the parameter b.

The hyperplane $h(\mathbf{x})$ can be reformulated in terms of the α_i parameters as follows:

$$h(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b. \tag{2.29}$$

Unfortunately, data are in general not perfectly separable by an hyperplane and the optimization problem has to be adapted in consequence. We thus talk about the *soft-margin* SVM and we refer to Figure 2.5 for an illustration of this situation.

The soft-margin SVM consists in the introduction of slack variables which measure discrepancies between data points and the margin. These variables will enable to find the best hyperplane separating the data but allowing simultaneously some data points to lie on the wrong side of the hyperplane or inside the margin.

Mathematically, the formulation of the soft-margin SVM is as follows

$$\min_{\mathbf{w},b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i \tag{2.30}$$

$$\text{subject to } \xi_i \geq 0 \text{ and } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \ \forall i, \tag{2.31}$$

where C is a now a parameter of the problem. A small value of C will be more permissive with respect to misclassified samples and will correspond to a larger margin while a high value of C will restrict at most the misclassification errors et the expense of the margin.

Solving this optimization problem leads to the same equation for the hyperplane $h(x)$ as in Equation (2.29) with the weight vector $\mathbf{w}$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i. \tag{2.32}$$

The samples x_i that contribute to the predictive model are the one for which α_i is non zero and these samples are also called the *support vectors*. Instances for which $\xi_i = 0$ are such that $0 < \alpha_i < C$ and they are on the boundary of the margin while observations i with $\alpha_i = C$ are inside the margin and can thus be correctly or incorrectly classified depending on the side of the hyperplane they are.

In Equation (2.29), the inner product $\mathbf{x}_i^T\mathbf{x}$ is called a linear kernel and is noted $K(\mathbf{x}_i, \mathbf{x})$. This kernel function measures the similarity between $\mathbf{x}_i$ and $\mathbf{x}$ by a simple product but can be replaced by other non linear kernel functions to adapt SVM to non linear classification.

2.4.2 Interpretability

Like tree ensemble methods, SVM is a method frequently used because of the interpretable results it provides. Indeed, the solution of the optimization problem is a weight vector $\mathbf{w}$ (see Equation 2.32) associating a certain importance value in the prediction to each input feature.

Absolute values of these weights are thus often used to obtain intuition about the most important variables. Similarly to tree based importance scores, absolute weights provide a ranking of the features.

Furthermore, the sign of the weights is useful to collect information about the involvement of a variable in one class or the other one. For instance in a healthy/diseased classification, this could be helpful to highlight a pattern specific to the diseased patients. Although it provides intuition about the features responsible for a phenotype, one cannot consider the weight map as a "univariate statistical map" and threshold it to provide the "contributing voxels".

Many papers in the biomedical field have based their work on the use of SVMs because of their simplicity, their interpretability, and their accuracy in high dimensional settings. In particular, in the neuroimaging field, many works mainly based on Support Vector Machines have arisen since last decade [Mourão-Miranda et al., 2005, LaConte et al., 2005, Orrù et al., 2012] to cite but a few.

2.4.3 Other linear methods

We mention here few other linear methods well known for their good performance in terms of accuracy and interpretability.

Multiple kernel learning Multiple Kernel Learning (MKL) has been proposed by Bach et al. [2004], Rakotomamonjy et al. [2008] and consists in considering the linear kernel $K(\mathbf{x}_i, \mathbf{x})$ as a linear combination of several kernels, i.e.

$$K(\mathbf{x}_i, \mathbf{x}) = \sum_{i=1}^{M} d_i K_i(\mathbf{x}_i, \mathbf{x}), \tag{2.33}$$

with $d_i \geq 0$ and $\sum_{i=1}^{M} d_i = 1$. Each kernel K_i is thus associated to a subset of features, which can correspond to different regions of the brain or different imaging modalities for instance. The solution of the optimization problem thus attributes simultaneously a weight d_i to each kernel and a weight w to each feature value if kernels K_i are linear kernels. Absolute values of the weights d_i denote the importance of each subset of features in the classification function. Sparsity in the kernel weights is enforced by the L_1 constraint on d_i, i.e. some weights can be null and the corresponding subset of features is not contributing to the model.

Lasso Tibshirani [1996] proposed a linear regression method with L_1 penalization of feature weights. The model $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ obtained is the result of the following optimization problem

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{w}^T\mathbf{x}_i - b)^2 + \lambda \sum_{i=1}^{m} |w_i| \right\}. \tag{2.34}$$

Through the $\mathcal{L}_1$ penalization of the weights, such procedure enforces sparsity of weight vector and thus embeds variable selection. For classification problems, the Lasso penalty can be applied to the *Logistic Regression* algorithm giving rise to the following formulation

$$\max_{\mathbf{w},b} \left\{ \sum_{i=1}^{n} \left[(y_i(\mathbf{w}^T\mathbf{x}_i + b) - \log(1 + e^{\mathbf{w}^T\mathbf{x}_i + b}) \right] - \lambda \sum_{i=1}^{m} |w_i| \right\}. \tag{2.35}$$

Group Lasso The link between Lasso and Group Lasso is similar to the one between SVM and MKL, i.e. Group Lasso is a version of Lasso taking into account groups of features. Such method can be convenient in situations where variables are organized in groups, as genes in a same biological pathway or voxels in a same brain region. This method has been firstly introduced in [Yuan and Lin, 2006]. The penalty is applied at the group level, such that the optimization problem in regression is as follows

$$\min_{\mathbf{w},b} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i - b)^2 + \lambda \sum_{g=1}^{G} \sqrt{m_g} \parallel w_g \parallel_2 \right\}. \tag{2.36}$$

In this formulation, there are G groups, the cardinality of group g is m_g and its coefficient vector is w_g. As $\parallel w_g \parallel$ is null only if all its components are null, this penalization enforces sparsity between groups. A Logistic Regression adaptation for classification has been proposed by Meier et al. [2008]. We can also cite other adaptations of group lasso; sparse group lasso [Friedman et al., 2010], overlap and graph group lasso [Jacob et al., 2009], fused lasso [Tibshirani et al., 2005], among others.

Chapter 3

Machine learning in neuroimaging

Chapter overview

This chapter provides the neuroimaging background of this thesis. Image preprocessing and machine learning state of the art in this field will thus be discussed here. We first describe in this chapter what is brain imaging and how we deal with this type of data. In Section 3.2 we provide a non exhaustive list of interesting publications made in the field of machine learning for neuroimaging, in particular for computer aided diagnosis systems for Alzheimer's disease. We finish the chapter by describing the data we will make use of in our experiments.

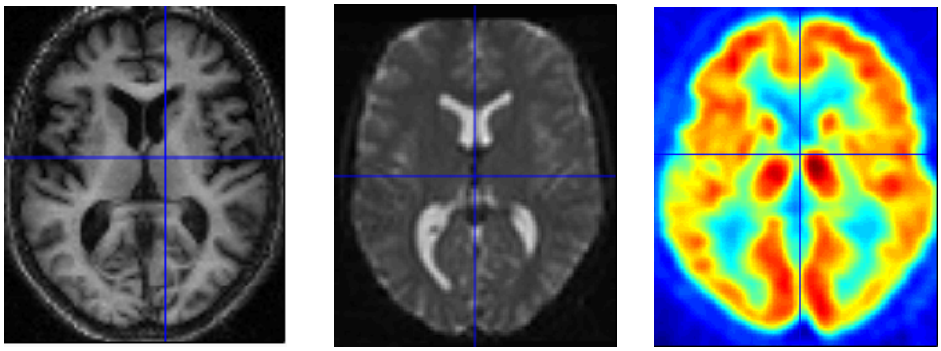
3.1 Principles of neuroimaging

The target of this section is the introduction to basic concepts of the neuroimaging field. In particular, we describe here the different imaging modalities used later in the manuscript, but also their pre-processing and the classical statistical univariate approach for inference.

Figure 3.1 illustrates three types of modalities widely used: structural and functional MRI, and PET imaging. The type of modality chosen for a study depends on what you are interested in.

Indeed, structural imaging will provide information about the brain anatomy and could be helpful to localize a lesion for instance. Volume measures of brain regions can also bring information about atrophy/hypertrophy linked to cognitive behaviours or disease. By comparison functional imaging aims at the characterization of brain activity. Functional MRI (fMRI) data are generally used to study the link between brain activity and cognitive task through the "blood-oxygenation-level dependent" (BOLD) signal. Finally PET images reflect metabolic processes depending on the radiotracer used. Medical diagnoses of neurodegenerative diseases or brain tumour detections are often carried out with this imaging modality.

We describe these modalities in Subsection 3.1.1. In Subsection 3.1.2, we provide a small overview of preprocessing techniques necessary for the analysis of neuroimages. This stage is in particular mandatory before any multi-subject analysis because brain images have to be made comparable across the group of subjects.



(a) Structural MRI. (b) Functional MRI. (c) PET scan.

Figure 3.1 – Different brain modalities providing information about the brain anatomy (Fig.3.1(a)), neuronal oxygen consumption variation and thus its activation (Fig.3.1(b)), and metabolism (Fig.3.1(c)).

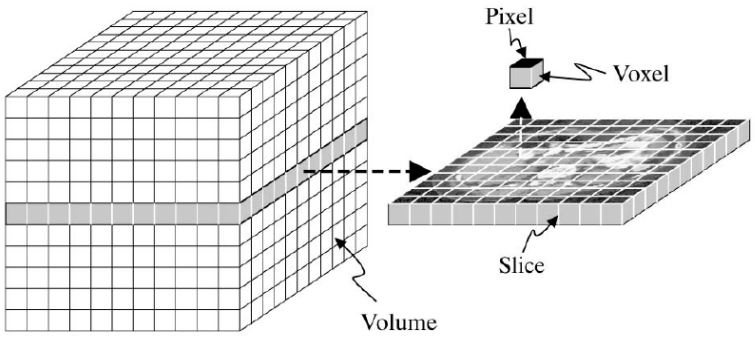


Figure 3.2 – Matrix structure of a neuroimage. Image taken from [Prince and Links, 2006].

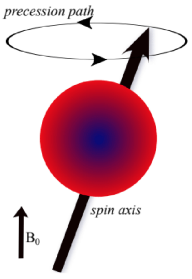
3.1.1 Neuroimaging modalities

We describe in this subsection two different acquisition techniques: Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET). These are the two types of imaging modalities used for experiments in this thesis.

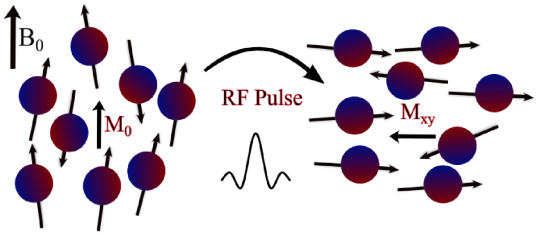
After acquisition, images are found in the form of a matrix with three dimensions (x, y, z), as illustrated in Figure 3.2, or four (x, y, z, t) dimensions in the particular situation of time series data. Images are acquired by slice of 2D images or by full 3D reconstruction. As a bi-dimensional image is divided into pixels, 3D brain images are divided into volume elements, called *voxels*. Each element encodes information about the structure or the activity at a location in the brain.

The number of voxels describing the brain depends on the spatial resolution and field of view chosen at the acquisition time. The image quality is influenced by this parameter but also by other factors like contrast and, importantly, signal to noise ratio.

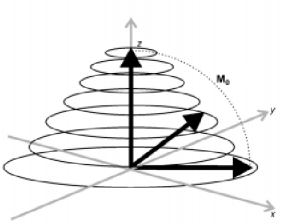
For machine learning use, inputs are in general organised as a unidimensional vector. It is thus obviously important to keep track of the relation between the components



(a) Magnetic moment and precession of a nucleus under B_0.



(b) Effect of the RF pulse application.



(c) Phenomen of relaxation.

Figure 3.3 – Magnetic Resonance Imaging. Images taken from [Möllenhoff, 2016].

of this vector and the spatial coordinates of the corresponding voxels in order to be able to interpret the results.

Magnetic Resonance Imaging

Structural Magnetic Resonance Imaging (sMRI) is an imaging technique based on the principle of nuclear magnetic resonance, a property exhibited by some nuclei which absorb and emit an electromagnetic radiation when they are under the influence of a magnetic field. For medical imaging, the nucleus of interest is the hydrogen nucleus ^1H (a single proton). This is the most frequent type of atom present in biological tissues. Such nucleus behaves like a small magnet with two possible energy states (spin $\pm\frac{1}{2}$).

The MRI scanner is composed of a large magnet that produces a strong and static magnetic field B_0 (> 1 Tesla). In the presence of this magnet, the nuclei, which were initially randomly oriented, align their spin in a parallel or anti-parallel way with the magnetic field. Parallel state corresponds to the lowest state of energy and anti-parallel to the highest one and the difference of energy between the two levels is given by $\Delta E = \gamma \bar{h} B_0$. The movement of precession is now dependent on the magnetic field B_0. In particular, protons are precessing at a frequency proportional to B_0, called the *Larmor* frequency. Placing tissues in the B_0 field thus magnetizes its atoms allowing the measurement of the ^1H nuclei resonance properties. This stage of magnetization is illustrated in Figure 3.3(a).

To obtain a measurable signal from the tissues, an oscillating magnetic field B_1 of

lower intensity ($\approx 50mT$) and of frequency equal to the protons precessing frequency, i.e. the *resonance* frequency, is applied. Microscopically, the RF pulse brings energy to the spins and longitudinal magnetisation is decreasing. The spins finally acquire the highest state of energy. Moreover, spins are now precessing in phase, leading to a net transverse magnetization M_{xy}. The RF perturbation is illustrated in Figure 3.3(b). Macroscopically, the net magnetization is precessing about B_1 with a frequency γB_1.

When the pulse is interrupted, protons progressively return to their initial state with their initial magnetisation M_0 under the influence of B_0. This phase is named the *relaxation* and is represented in Figure 3.3(c). The net transverse magnetization is a measurable signal. Macroscopically, the transverse component of magnetic moments is decreasing while the longitudinal one is increasing to finally come back to their initial stage under the influence of B_0. These relaxation phenomena are characterized by an exponential decrease/increase. More precisely, T_1 is called the longitudinal relaxation time and corresponds to the duration after which the longitudinal magnetisation has recovered about 63% ($= 1-\frac{1}{e}$) of its initial value. Similarly, T_2 characterizes the transversal relaxation and corresponds to the duration after which the transversal magnetisation has lost about 63% of its value during the application of B_1. The time constants T_1 and T_2 are different across tissues and these differences enable the construction of contrast images (grey and white matter distinction).

Positron Emission Tomography

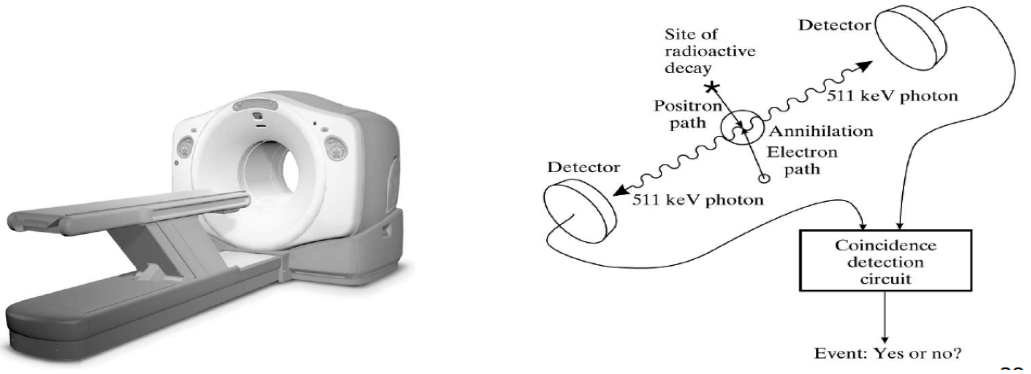
Positron Emission Tomography (PET) is a nuclear medical imaging technique useful to measure positron emissions from a radio-tracer injected in the body. In brain imaging, the most frequently used radio-tracer is the fluorodeoxyglucose, enabling the measure of neuronal glucose consumption. Measures obtained with PET imaging mostly reflect energy intake which is a characteristic tightly linked to brain activity.

For image acquisition, the first stage consists in injecting a radio-tracer intravenously in the patient. A radio-tracer is a biological component (like glucose) that has been marked by a radioactive atom in order to be able to follow its consumption in the body. There is a waiting time for the molecule to fix in the body, then the patient can be placed in the scanner (illustrated in Figure 3.4(a)) in order to record radiations emitted by the radioactive marker.

The radioactive decay causes the emission of a positron which, after a small path of a few millimetres, meets an electron and they annihilate themselves. As a consequence, two photons (with an energy of 511keV) are emitted on the same line but in opposite direction. If both photons are detected, then the coincidence event is counted and used for image reconstruction, as illustrated in Figure 3.4(b). With a sufficiently large number of events detected, it is then possible to reconstruct the spatial concentration of annihilation sites, therefore mapping in the volume of the brain the distribution of the radiotracer.

3.1.2 Image preprocessing

In this thesis, these imaging modalities are analysed with machine learning methods. In particular, we study the problem of patient classification according to their MRI or PET scan. The input matrix of such a problem is thus composed of lines corresponding to distinct patients and columns representing each a different voxel of the acquired brain image.



(a) PET scan machine. (b) Principle of radiation emission.

Figure 3.4 – Positron Emission Tomography. Images taken from [Prince and Links, 2006].

However, when image acquisition is performed, each patient is not necessarily imaged with the exact same position. Furthermore, each patient has a brain of specific size and shape. For these reasons, before any machine learning analysis, brain images have to be spatially normalized in a common reference space to make them comparable between each other. After this operation, each voxel or each (x, y, z) location refers to the same anatomical brain point in a typical reference space for every patient of the database.

In more details, this normalization operation consists in resizing and reshaping each brain image with respect to a brain image template in such a way that every structure of the brain is described at the same point for each patient. Images are generally brought into the MNI (Montreal Neurological Institute) reference space [Mazziotta et al., 2001]. This helps to report the results from statistical analyses. As normalization does not completely correct spatial anatomical variability between individuals, a smoothing procedure is generally applied to the images after normalization. It consists in the application of a spatially stationary Gaussian filter for which the *full width half max* is a parameter depending in general on the modality. Essentially, the filter averages a voxel value with values in its neighbourhood and it increases the signal to noise ratio.

In our work, we perform these preprocessing stages with the Statistical Parametric Mapping (SPM) toolbox [1] [Penny et al., 2011a]. This toolbox also enables *univariate* statistical image analysis.

3.1.3 Univariate per voxel analysis

Univariate analysis, in opposition to multivariate methods, only involves the statistical analysis of one variable at a time. Typically in neuroimaging, the idea is to link the signal in each individual voxel location across all the subjects with some explanatory variables, for example the age of the subject and/or their category (such as diseased vs. healthy). Then the statistical significance of this link is assessed. Since the same operation is performed for every voxel of the brain, a p-value is attributed to each voxel and so a statistical map is constructed. This map is then thresholded (e.g. a threshold of $p < 0.001$ or $p < 0.05$ is chosen) to highlight only the statistically significant voxels for the condition analysed, while accounting for the multiplicity of tests. For example,

[1] http://www.fil.ion.ucl.ac.uk/spm/

when comparing two (or more) groups of patients, the map shows the brain regions with significantly different signals.

More precisely, a statistical parametric analysis relies on a *General Linear Model* (GLM) approach, where explanatory variables are linearly combined to describe the signal at hand [Friston et al., 1994]. Statistical tests are performed afterwards by building contrasts of interest, i.e. a linear combination of model parameters, and inferring their their significance through a t- or F-test.

Let us assume a study with n samples. For instance, these samples correspond to a multi subject study (one image for each patient, n patients) and two groups of subjects are compared. Mathematically, the $n \times 1$ signal vector Y at a given voxel is modelled like this

$$Y = X\ \beta + \epsilon \tag{3.1}$$

with X the design matrix, β the parameters that will be estimated by a least-squares regression and ϵ the residual noise. Matrix X has n rows and m columns and each column encodes a different design factor. More exactly, the design matrix defines all the experimental conditions which can have influenced the voxel signal. It therefore includes the variables of interests (those we assume to influence the signal, such as the effect of a stimulus or the membership to different groups of patients) but also the potential confounders (i.e. factors that can influence the measured signal but are different from the experimental conditions of interest). The error ϵ models the residual variability not explained by the experimental and confounding effects.

As each statistical test is repeated for all voxels of the brain, the problem of multiple comparisons arises and corrections have to be applied to control for the risk of false positives at the level of the whole brain. The Bonferroni correction [Dunn, 1961] is a straightforward and simple solution but is too conservative when the observations are not independent, as it is the case with (smooth) brain images. An alternative is to use random field theory which would be more appropriate to neuroimaging data (see [Brett et al., 2003] for more details).

3.2 Machine learning for neuroimaging

In this section, we provide a non exhaustive state of the art of machine learning methods and approaches proposed for exploiting neuroimaging datasets in general but also in the particular case of Alzheimer's disease.

3.2.1 General overview

One of the most popular machine learning algorithms used in the neuroimaging field is the Support Vector Machines (SVM) [Hearst et al., 1998]. Often used in its linear version, it has proven its good behaviour with high dimension (m) and small sample (n) problems [Mourão-Miranda et al., 2005, Magnin et al., 2009, Orrù et al., 2012].

Another method that is known for its state-of-the-art performances on machine learning problems with large m/n ratio is the Random Forests method [Breiman, 2001]. Although less usual in the neuroimaging community than SVM, it has however provided good results in the field with fMRI for instance [Langs et al., 2011, Richiardi et al., 2010].

For a few years now, the machine learning field has observed the emergence of deep learning methods [Goodfellow et al., 2016] and some applications of these methods

have appeared in the neuroimaging literature. For example, Plis et al. [2014] applied deep learning methods on different structural and functional brain imaging datasets. Datasets used in this work are typically composed of hundreds to thousands of samples. They showed good promise for the use of deep learning in classification tasks with neuroimaging. In [Shen et al., 2017], the authors provided a review of deep learning approaches for the analyses of medical images, from image registration to computer aided diagnosis system. Although such methods seem to be very promising in terms of predictive performance, deep neural network models remain difficult to interpret [Suk et al., 2014].

In addition to the classification algorithm as an entity, a classification framework can be composed of other components: feature extraction (or feature engineering), feature selection, or dimensionality reduction. Methods based on feature vectors directly extracted from the images are called *voxel-as-feature (VAF) based method*. Classifiers based on neuroimages require often one or more feature reduction techniques as the number of brain voxels extracted from one neuroimage is huge (from $50,000$ to $300,000$ features) and the number of samples is low (from 50 to a few hundreds). It is thus difficult for a machine learning algorithm, as efficient as it could be, to reliably identify very discriminative features when there are so few instances and so many possible explanatory variables.

For MRI data, there is no interest to work directly with voxels and feature extraction is a necessary operation. Especially, information about regional volumes and shapes or tissue densities are computed from the MRI. Brain atrophy is thus estimated with density maps of grey matter, which are provided by voxel-based morphometry methods [Ashburner and Friston, 2000]. Classifiers can thus be learnt either using the map as a whole or by performing feature selection. Other common features to distinguish cases based on their brain atrophy are volume and shape measure in the Hippocampus. Unlike MRI, PET data can be directly exploited in a classification algorithm to estimate information included in a dataset.

To avoid a feature selection procedure, some works directly select brain regions already identified in previous literature as relevant for the disease. However, it prevents the discover of new regions of interest. Among feature selection procedures, we can distinguish three big categories of methods: the *wrapper* methods, the *filter* methods and the *embedded* methods [Guyon and Elisseeff, 2003]. Wrapper methods are computationally demanding as they compute classifier accuracy for all possible (or many) subsets of features. This is not achievable with the number of variables in neuroimaging data. Filter methods are independent on a classification algorithm and eliminate features based on a correlation criterion or other similar measure able to quickly highlight features having no link at all with the output. Embedded methods include in their machine learning algorithm a feature selection process, like LASSO or CART decision tree [Tibshirani, 1996, Breiman et al., 1984].

Finally, dimensionality reduction techniques are commonly used in machine learning with neuroimaging. However they reduce the interpretability of the classifier. Among these feature reduction methods, we can notably cite Principle component analysis (PCA) or Partial least square (PLS) [Jolliffe, 1986, Geladi and Kowalski, 1986].

3.2.2 Computer aided diagnosis for Alzheimer's disease

For several years, the interest of neuroscientists for leveraging machine learning in order to support disease diagnosis has never stopped to grow. Indeed, for medical diagnosis, a system based on machine learning could be helpful to make a decision about

a patient for which a medical doctor would not be confident about the possible diagnosis. Diagnosis systems extract knowledge from an original dataset representative of the problem and use this knowledge to generalize and infer a decision about a new example. They perform group analysis, detect similarities between subjects in a same group, and use these observations to classify new data. However, as a doctor would motivate the diagnosis he would have provide based on some medical criteria, diagnosis systems should be interpretable in order to verify the consistency of the diagnosis advised by the machine. The most popular methods in machine learning for neuroimaging are thus those enabling interpretations about the problem, such as the brain regions discriminating two classes of patients.

Much research has already been undertaken in this topic for Alzheimer's disease (see [Rathore et al., 2017] for a review of the main publications in this field). The modalities used are structural imaging [Klöppel et al., 2008, Cuingnet et al., 2011] or functional imaging [Gray et al., 2012] like PET data. In [Klöppel et al., 2008], authors used SVM to classify grey matters from T1-MRI images of pathologically proven AD patients and cognitively normal (CN) individuals. They obtained classifiers of about 90% accuracy using whole brain images. Cuingnet et al. [2011] used T1-weighted MRI from the ADNI[2] database, which is a dataset publicly available including about $1,000$ subjects from CN, MCI and AD classes. They investigated three classification problems: AD vs. CN, CN vs. mild cognitive converters (MCIc) and MCIc vs. MCIs (stable MCI). Their classification frameworks were mostly based on SVM but with diverse features (voxel-based or regional volume information). They obtained good results only for the classification AD vs. CN. In their work, Gray et al. [2012] notably used FDG-PET imaging in combination with clinical information and MRI data from the ADNI database to build a classifier able to distinguish AD and CN individuals at an accuracy of 88%. More precisely, they extracted regional volume information from MRI images recorded at the baseline and after 12 months follow up and also computed FDG-PET intensities per mm^3 for each region.

There actually exists much more computer aided diagnosis (CAD) systems designed for the classification between AD and cognitively normal people than for AD vs. MCI patients. However the aid of a computer is generally unnecessary for the distinction between CN and AD while differences can be less obvious between MCI and AD patients. Moreover, another big challenge is to be able to predict the evolution of a MCI patient. Nevertheless, there are few databases available to handle this problem, as people are often diagnosed very late in the disease progression. We provide here below some examples of research work for the classification of AD and MCI patients or the prognosis of MCI patients to converter or non converter.

Sevaral studies have investigated the benefits of the combination of multiple information. Simple feature concatenation does not lead always to an efficient classifier. Feature reduction can help to improve classifier performance. For instance, Zhang et al. [2012] have combined MRI, FDG-PET and cerebrospinal fluid (CSF) information for the distinction between AD, MCI and control patients but also to predict evolution from MCI to AD disease stage. They used an atlas to obtain tissues volumes in brain regions from MRI data and they averaged PET values per brain area. Suk and Shen [2013], Suk et al. [2014] proposed a deep learning approach to obtain high-level features from MRI and PET data. Using such features, their classifiers obtained very high efficiency. In particular, [Suk et al., 2014] obtained more than 95% of accuracy for the classification of AD vs. CN, 85% of accuracy for the classification of MCI vs. CN and less than 75% of accuracy for the classification of MCI converters vs. MCI stable patients. In

[2] http://www.adni-info.org

Table 3.1 – Demographic details of each dataset and chapter in which the dataset is used. μ and σ stand for average and standard deviation respectively.

		Sex			Age			Chapter
		#	M	F	μ	σ	Range	
CRC	MCI	23	14	9	73.43	7.80	58-84	4, 6 & 7
	MCIc	22	12	10	75.64	4.61	67-82	
OASIS	CN	50	22	28	75.00	6.70	60-92	4, 6 & 9
	AD	50	22	28	75.30	6.80	61-96	
CRC_2	AD_t	22	14	8	77.73	8.33	58-93	8
	AD_{TPJ}	30	17	13	78.90	6.71	59-90	
ADNI	AD	207	121	86	74.79	7.97	55-90	8
$ADNI_2$	AD	94	56	38	75.60	7.32	55-88	9
	MCI	106	71	35	75.76	7.50	57-89	

[Casanova et al., 2013, Segovia et al., 2014], clinical scores were used in combination with imaging modalities (MRI in the first one and PET for the second one) also for the construction of prognosis system. Casanova et al. [2013] only concatenated imaging matrices (grey matter, white matter and CSF) and cognitive scores while Segovia et al. [2014] investigated two different dimension reduction methods (PCA and PLS) for brain images before simple combination with clinical scores.

Moreover, Moradi et al. [2015] and Gray et al. [2013] have both worked on computer aided prognosis system for the prediction of MCI to AD conversion with Random Forests algorithm. The former used Random Forests as the final classifier of concatenated data (MRI features and cognitive scores). In the latter work, Random Forests method is also used to compute similarity measures of input features for feature selection. Both studies are based on the ADNI database. In this database, some of the subjects have been followed during several years and many imaging modalities and clinical scores are available for each subject.

3.3 Datasets

In this section, we describe the datasets studied in the next chapters. The study of these datasets follows two main goals. On the one hand, we study tree based ensemble methods and their extensions with small neuroimaging datasets for Alzheimer's disease. Methods studied and/or developed in this thesis are however generalizable to other neuroimaging problems. On the other hand, we are interested in acquiring a good understanding and possibly new information about Alzheimer's disease thanks to machine learning methods applied on these datasets.

Demographic details of all datasets are provided in Table 3.1. As preprocessing has been achieved once for each dataset, we also provide the preprocessing stages applied to each dataset in this part of the manuscript.

CRC Data - MCIc vs. MCIs This dataset is useful to deal with the prognosis of Alzheimer's disease. Medical doctors are currently not able to claim with brain imaging

if a MCI patient is susceptible or not to develop the disease. Machine learning algorithms could find differences between patients who convert later and those who stayed stable while a human would encounter difficulties to generalize and to detect the differences.

In particular, 45 patients presenting mild cognitive impairment were enrolled in a longitudinal study achieved by the Cyclotron Research Centre (University of Liège, Belgium). Patients were selected based on Petersen's criteria [Petersen and Negash, 2008] for MCI, including memory complaints, objective memory deficits on neuropsychological testing, no evidence of global cognitive decline and preserved activities of daily living. At the beginning of the study, one Fluorodeoxyglucose (^{18}F-FDG) positron emission tomography (PET) image was recorded for each patient. During the next four years, patients were followed and evaluated repeatedly with neuropsychological tests. Conversion was detected as soon as a patient fulfilled the diagnosis criteria for Alzheimer's disease at a follow-up assessment, that is, objective deficit in more than two cognitive domains, general cognitive decline and significant reduction of autonomy in everyday life activities. Along the time of the study, several individuals converted from MCI to Alzheimer's disease and, at the end of the study, the total number of converters (MCIc) was 22. Demographic details about patients at their entrance in the study are reported in Table 3.1.

As required, the protocol of the study was accepted by University Ethics Committee in Liège. All patients received a written and oral description of the study and then provided a written consent. Concerning the acquisition of the images, they were performed 30 minutes after injection of the ^{18}F-FDG radiopharmaceutical, by means of a Siemens ECAT HR+ PET gamma camera (3D mode; 63 image planes; 15.2cm axial field of view; 5.6mm transaxial resolution and 2.4mm slice interval). Images were reconstructed using filtered backprojection including correction for measured attenuation and scatter using standard software.

After acquisition, images were pre-processed using SPM8. All PET images were spatially normalized to the MNI reference space using the template matching approach implemented in SPM8 [Ashburner et al., 1999, Penny et al., 2011b]. Spatial normalization was followed by an intensity scaling by cerebellar uptake, with the cerebellum delineated according to the automated anatomical labelling (AAL) atlas [Tzourio-Mazoyer et al., 2002]. To finally obtain a feature vector for each patient, a mask was applied to extract only the voxels included inside the brain volume. This stage gave rise to a feature vector composed of a little bit less than 220,000 variables per image.

OASIS Data - AD vs. CN This dataset is a good example of structural MRI data for AD. Although it is generally easy to make differences between AD and CN patients with sMRI (much more severe atrophy), AD patients considered here are not in an advanced stage of the disease. For this reason, the task should not be as easy as it could be with severe AD.

More precisely, we carry out experiments in this manuscript on images from the Open-Access Series of Studies (OASIS[3]) [Marcus et al., 2007]. We use structural MRI from demented (50) and non-demented (50) old individuals, age and gender matched. The fifty demented subjects studied here correspond to the old people diagnosed, by using the Clinical Dementia Rating (CDR) scale, very mild ($CDR = 0.5$) to mild ($CDR = 1$) Alzheimer's disease. Data were pre-processed using SPM8. All the repeats for each session were averaged and then the grey matter was segmented and normalized. Furthermore, grey matter images were smoothed with a [8 8 8] mm full width at half maximum (FWHM) Gaussian kernel. Finally, only voxels with a probability of being grey matter greater or equal to 30% in all subjects were selected ($\simeq 320,000$ voxels).

[3]http://www.oasis-brains.org

CRC_2 dataset - Typical vs. TPJ The Cyclotron Research Centre provided us a dataset composed only of Alzheimer's disease patients. Researchers visually detected two different metabolic profiles among all AD patients and they separated them in two classes. For these patients, some clinical scores were not assessed whereas they would have been useful in order to interpret which could be the clinical differences between these groups. We thus propose to use some ADNI data in order to infer more information about individuals from the CRC.

The dataset provided by the CRC is composed of fifty-two FDG-PET images. These images were processed by first spatially normalising them with a template available in SPM. Then an average template was created in order to normalize original images according to a new template representative of the database. Intensity normalisation by cerebellar mean intensities was subsequently performed. By these operations, feature vectors of about $220,000$ voxels were also obtained.

ADNI dataset - Typical vs. TPJ In order to characterize Alzheimer's disease, we decide to work with a public dataset, the ADNI database, as stated above. Such database contains a lot of neuropsychological information for each patient. We used 207 FDG-PET images of AD patients.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a public-private partnership launched in 2003. The principle aim of this initiative has been to investigate whether serial magnetic resonance imaging (MRI), positron emission tomography, other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment and early Alzheimer's disease. For more information, see `www.adni-info.org`. AD patients enrolled in the ADNI study have obtained a score between 20-26 (inclusive) at the Mini-Mental State Exam. They have also exhibited a memory complaint and an objective memory loss measured by education adjusted scores on Wechsler Memory Scale Logical Memory II. Moreover, they have shown a clinical dementia rating (CDR) of 0.5-1 and have satisfied the NINCDS/ADRDA criteria for probable AD.

Images used for this work were pre-processed before using them for machine learning. Spatial normalization with the average template was followed by intensity normalization, similarly to the CRC_2 dataset. Final resolution of the ADNI images are also the same as these images, giving rise to feature vectors of approximately $220,000$ variables.

$ADNI_2$ dataset - AD vs. MCI Part of data used in Chapter 9 consists of a dataset including 106 ^{18}F-FDG PET images for patients with mild cognitive impairment and 94 ^{18}F-FDG PET images of AD patients. They were obtained from the ADNI database and correspond to images recorded at their entrance in the study. We analyse with this dataset the evolution of metabolic patterns through disease stages.

MCI patients enrolled in the ADNI study have respected some eligibility criteria. In particular, they have obtained a score between 24-30 (inclusive) at the Mini-Mental State Exam, they have exhibited a memory complaint and an objective memory loss measured by education adjusted scores on Wechsler Memory Scale Logical Memory II. Moreover, they have shown a CDR of 0.5, absence of significant levels of impairment in other cognitive domains, essentially preserved activities of daily living, and an absence of dementia. AD patients enrolled in the ADNI study have obtained a score between 20-26 (inclusive) at the Mini-Mental State Exam. They have also exhibited a memory complaint and an objective memory loss measured by education adjusted scores on Wechsler Memory Scale Logical Memory II. Moreover, they have shown a clinical dementia rating (CDR) of 0.5-1 and have satisfied the NINCDS/ADRDA criteria for probable AD.

Images were spatially normalized to MNI space and then normalized in intensity according to cerebellum mean activity, giving rise to feature vectors of approximately $220,000$ variables.

Part II

Tree ensemble methods in neuroimaging

Chapter 4

Tree ensemble variable importances in high dimension

Chapter overview

In this chapter, we analyse, theoretically and empirically, some properties of the MDI variable importance scores provided by tree ensemble methods in the very high dimensional setting commonly faced in neuroimaging datasets. We firstly derive, theoretically, the minimal number of trees that should be built in a forest to have seen each and every feature at least once, highlighting how large this number can be in typical neuroimaging datasets. Secondly, we perform an empirical analysis of the stability of importance scores as a function of the main method parameters. We conclude the chapter by making some recommendations based on our theoretical and empirical investigations.

4.1 Introduction

In Random Forests, as well as in Extremely Randomized Trees, the best variable selected to split each node in the decision or regression trees of the forest is evaluated among only K features randomly drawn (locally) without replacement [Breiman, 2001]. The splitting feature is the one giving rise to the largest reduction of the impurity measure. Consequently, it is possible that some considered features obtain a null importance after having been evaluated and are thus never chosen in the forest. We can interpret such a situation by the fact that there were always other features considered as better at describing the relationship between X and Y than such variables. This can happen when they are evaluated at the root node but also later in the tree, if they do not provide enough information when they are joined to the other variables selected for the parent nodes. As a consequence, the fact that a variable has a zero (or small) importance does not imply that it is not relevant for predicting the output. This kind of effect is called a masking effect and is discussed more formally in [Louppe et al., 2013]. The larger K, the more important this effect will be.

In the case of an input matrix of very high dimension ($m \gg n$) and a too small tree ensemble, a zero importance score could however also have been attributed to some features simply because they have never been even considered for node splitting during the whole learning process. Intuitively, if the total number of test-nodes of the ensemble is small compared to m/K, it is not unlikely that many of those features having a zero

importance value have actually never been considered during the ensemble training process. Since the number of test nodes of each tree in the ensemble is upper bounded by the sample size n, the smaller the sample size, the more likely this is to happen for a given number of trees. To quantify the importance of this phenomenon, in particular in neuroimaging datasets, we analyse theoretically in Section 4.2 the number of trees that are required on average to have seen each input variable at least once, as a function of problem (m and n) and method (K) parameters. As we will see, this problem may be reduced to an already solved combinatorial problem. This first analysis will provide a lower bound on the number of trees required for a given problem to be able to attribute zero importance scores to masking effects only.

A further concern in high dimension is the stability of the importance scores as provided by an ensemble of a given size T. Ideally, to yield stable importance scores, each feature should actually be considered multiple times during the ensemble growing process. In general, one can thus expect that more trees than predicted by the previous theoretical analysis will be needed for variable importance scores to reach convergence. In the second part of the chapter, we will examine this question via an empirical simulation study on both artificial and real neuroimaging datasets. We will in particular analyse the impact of both model and data randomizations on stability as measured through several criteria and depending on method parameters K and T.

Related works

Several previous works about Random Forests have already taken interests in analysing the amount of trees that should be fitted in a forest, e.g., [Latinne et al., 2001, Oshiro et al., 2012, Genuer et al., 2010]. Most of these works however focus on the number of trees required to reach convergence in terms of predictive performance, not in terms of variable importance scores. For example, Latinne et al. [2001] proposed to use a McNemar test to decide whether increasing the size of an ensemble leads to a significant improvement of accuracy. They designed a procedure based on this test to determine the minimum number of classifiers to include in an ensemble without causing significant loss of accuracy. Oshiro et al. [2012] perform a similar study using the area under the ROC curve (AUC) as the main performance criterion and a Friedman test to assess the impact of an increase of ensemble size on AUC. More in relation to our work in this chapter, Oshiro et al. [2012] also studied the impact of ensemble size on the percentage of attributes used in the forest, as a function of dataset density (i.e., the ratio n/m). We will carry out a similar experiment in the second part of this chapter.

In [Genuer et al., 2010], the authors propose a strategy for variable selection, combining a prior variable ranking based on the mean decrease of accuracy (MDA) importance score and a stepwise variable elimination procedure based on out-of-bag error estimates. As a prelude to the introduction of this method, they carry out in their paper an empirical analysis of the impact of problem (n, m, and feature correlations) and method (K and T) parameters on MDA importance scores. These experiments are carried out on an artificial and a real (genomic) dataset, where average and standard deviations of importance scores over several Random Forests runs are reported for the most important variables. Similar experiments will be carried in the second part of this chapter with the MDI importance measure, both on one artificial problem and two neuroimaging datasets.

Saeys et al. [2008] defines the stability of a feature selection or ranking techniques as the modification of the output of these methods when small modifications are applied on the dataset. They highlight in their paper the importance of stability as a criterion to evaluate feature selection techniques and show that ensemble methods are very ef-

fective at improving stability. We will adopt the quantitative stability measures used in this work in the context of our empirical analysis in Section 4.3.

4.2 Combinatorial analysis

In this section, we want to estimate the number of trees necessary to build in a Random Forests or Extremely Randomized Trees ensemble in order to have observed each input variable at least once (on average). For this computation, a variable is considered as observed as soon as it is selected among the K variables randomly picked at at least one node during the ensemble construction. As discussed in the introduction, this number will give an indication of the very minimum number of trees that should be built, for a given problem and parameter setting, to have some minimum guarantee that zero importance features are due to some masking effect or to the irrelevance of the feature and not to the fact that not enough exploration has been carried out.

We first derive this number theoretically in Section 4.2.1. This objective is achieved by reducing the problem of having drawn each feature at least once over T trees of average complexity (number of testing nodes) N_t to the problem of having drawn each feature at least once over $N_t \times T$ trials. We then instantiate and discuss this number for typical random forest settings and neuroimaging dataset sizes in Section 4.2.2.

4.2.1 Theoretical derivation

The first stage of our derivation consists in estimating the number D of feature random draws required to have seen all the features at least once. This is obtained by making a parallel between our problem and the so-called coupon collector's problem. Then, we evaluate the average number of testing nodes N_t per tree in a forest. Finally, the number of trees T will be simply obtained by dividing the total number of draws D by N_t. We first focus on the case $K = 1$ and then extend the result to the case $K > 1$ [1].

Coupon collector's problem

Let us describe the problem in a more formal way. Basically, there are m different features and, each time a node has to be splitted, K feature(s) will be drawn randomly from the whole set of variables. Each feature has the same probability to be drawn. Equivalently, the problem is similar to considering a box of m different objects from which objects are sampled with equal probability and without replacement in group of size K and with replacement from one trial to another. We are interested in estimating the number of trials required to have drawn each object at least once.

In probability theory, such problem corresponds exactly to the *coupon collector's problem*. This problem is often motivated by the example of a kid who collects coupons in order to fill in an album with m different coupons. The coupons are purchased by pockets of $K \geq 1$ distinct picture(s) and we would like to know how many pockets need to be purchased on average to have completed the full album. All coupons have the same probability to be drawn. The *coupon collector's problem* has been studied extensively in the literature. It was first studied by De Moivre, Laplace and Euler in the XVIII$^{\text{th}}$ century (refer to [Stadje, 1990] for an historical description of this topic). They derived the probability $P(x)$ of having completed the collection after x pocket draws for $K \geq 1$. In 1930, a Hungarian mathematician named George Pólya formulated the expected value

[1] K should be an integer and is upper bounded by m. A common default value of K, in classification, is $\sqrt{m}$ rounded to the nearest integer.

of the waiting time necessary to complete the whole collection of coupons. Then further works followed for different variations of the original problem [Rosén, 1970, Stadje, 1990, Adler et al., 2003]. The expected number of sampling times when one single coupon is bought each time is also developed in probability books such as [Sheldon et al., 2002].

The *coupon collector's problem* is directly applicable to our problem of collecting the whole set of features, with the number of coupons and the size of the pockets corresponding respectively to the number m of features and the number K of features drawn at each node. The analysis presented in this first subsection has thus been extracted from previous literature and adapted using our own notations to our specific problem.

Proposition 1. *Let K be the number of features drawn without replacement at each trial and m the total number of distinct features, and let D denote the random variable representing the number of trials necessary for having collected all features. If $K = 1$, it follows from the classical coupon collector's problem [Holst, 1986] that the expected value of D corresponds to the following sum*

$$\mathbb{E}(D) = m \sum_{i=1}^{m} \frac{1}{i}. \tag{4.1}$$

Proof. D can be represented as a sum of random variables $D = d_1 + d_2 + \ldots + d_m$. In this sum, d_i is the random variable denoting the number of trials to acquire the ith feature when $i-1$ variables have already been collected. The acquisition of the ith feature represents a success and the behaviour of d_i can thus be modelled by a geometric distribution such that

$$d_i \sim Geom(p_i), \tag{4.2}$$

where p_i is the probability to draw an ith variable not seen yet, knowing that $i-1$ variables have already been collected. More details about geometric distribution can be found in [Sheldon et al., 2002, Wackerly et al., 2007]. When $i-1$ features have already been drawn, there remains $m-(i-1)$ possibilities among m to draw an unseen variable, given that each variable has the same probability to be picked. Mathematically, the parameter p_i of the geometric distribution is thus:

$$p_i = \frac{m-(i-1)}{m}. \tag{4.3}$$

The probability of drawing an ith variable after $k-1$ independent trials is given by:

$$P(d_i = k) = (1-p_i)^{k-1} p_i \tag{4.4}$$

and the expected value of d_i is $\mathbb{E}(d_i) = \frac{1}{p_i}$ [Wackerly et al., 2007, Sheldon et al., 2002].

We can thus easily obtain the expected value of the necessary number of trials D to have seen all the objects at least once. Indeed, D is simply the sum of the d_i for i from 1 to m. The average number of trials D is given by

$$\begin{aligned}\mathbb{E}(D) &= \mathbb{E}(\sum_{i=1}^{m} d_i) \\ &= \sum_{i=1}^{m} \mathbb{E}(d_i) \\ &= \sum_{i=1}^{m} \frac{m}{m-(i-1)} \\ &= m \sum_{i=1}^{m} \frac{1}{i}. \end{aligned} \tag{4.5}$$

□

In Equation (4.5), the sum of the reciprocals of the m first natural numbers is called the mth harmonic number H_m and its asymptotic value ($m \to \infty$) is such that:

$$H_m = \ln(m) + \gamma + \frac{1}{2m} + \mathrm{O}\left(\frac{1}{m^2}\right). \tag{4.6}$$

Injecting Equation (4.6) in Equation (4.5), we obtain:

$$\mathbb{E}(D) = m\ln(m) + \gamma m + \frac{1}{2} + \mathrm{O}\left(\frac{1}{m}\right), \tag{4.7}$$

where $\gamma \simeq 0.5772156649$ is the *Euler-Mascheroni* constant [Erdős, 1961, Holst, 1986].

Of interest is also how much D varies from one trial to another. Given that D is a sum of random variables from geometric distributions, an upper bound on its variance can be derived [Brayton, 1963, Doumas and Papanicolaou, 2012], which is as follows:

$$\mathrm{Var}(D) \leq \frac{\pi^2}{6} m^2. \tag{4.8}$$

The derivation of this bound is provided in Appendix A.

From this variance, it is then possible to derive a confidence interval for D using the Bienaymé-Chebyshev inequality [Sheldon et al., 2002].

Proposition 2. *Let $\mu = \mathbb{E}(D)$ and $\sigma^2 = Var(D)$ be respectively the mean and variance of the random variable D. The Bienaymé-Chebyshev inequality bounds the probability of D being in a certain interval as follows:*

$$P(|D - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{4.9}$$

or equivalently

$$P(|D - \mu| \leq k\sigma) \geq 1 - \frac{1}{k^2}. \tag{4.10}$$

Let us now consider the situation $K > 1$, which includes the common default setting $K = \sqrt{m}$. In this case, the average number of draws required on average can still be derived, although not in an easy-to-compute analytical expression.

Proposition 3. *Let $K > 1$ be the number of features drawn without replacement at each trial and m the total number of distinct features. The expected number of groups of K features that we need to draw in order to have seen the whole set of features is given by*

the following sum (proven in [Stadje, 1990, Sardy and Velenik, 2010, Ferrante and Frigo, 2012])

$$C_m^K \sum_{i=1}^{m-K} (-1)^{i-1} \frac{C_m^i}{C_m^K - C_{m-i}^K} + \sum_{i=m-K+1}^{m} (-1)^{i-1} C_m^i. \tag{4.11}$$

The derivation of Eqn. (4.11) is detailed in Appendix A.

This sum gives the average number of trials necessary for the machine learning method to have seen all features at least once during the construction of the forest. It is worth noting that, when $K > 1$, having seen all features does not mean that all features appear at a decision node of the forest, since only one feature is selected among K at each node. In contrast, when $K = 1$, there is almost[2] a one-to-one correspondence between the number of features seen and the number of features that appear at at least one node. Finally, let us notice that, when $K = m$, Equation (4.11) reduces to $\sum_{i=1}^{m} (-1)^{i-1} C_m^i$, which is equal to 1 as expected (by application of Newton's Binomial theorem).

Average number of testing nodes per tree

The complexity (i.e. the number of testing nodes N_t) of a tree is in general difficult to predict given only the size of the dataset. A dataset with small n will typically lead to smaller trees than a dataset with large n. However, the exact number of test nodes will depend on how easy it is, given the available features and method parameters, to separate the different classes with binary splits. Obviously, tree complexity is also influenced by the pre-pruning criterion used (e.g., the minimum number of instances per leaf), but in this work, we will assume that all trees are fully grown (i.e., the development of a branch is stopped either when all examples in the leaf are of the same class or all variables have a constant value) since this is the most common setting in the context of random forests. In the context of random forests, one should also take into account the fluctuations from one tree to another due to randomization and the impact of bootstrap sampling that reduces the sample size by 63.2% on average.

One can nevertheless obtain an upper bound on decision tree size, by using the following general proposition:

Proposition 4. *The total number of internal nodes N_i in a binary tree composed of N_l leaves is given by*

$$N_i = N_l - 1. \tag{4.12}$$

(The proof is easy to obtain by mathematical induction [Goodrich and Tamassia, 2008].)

In the worst case in terms of complexity, a fully grown decision tree has a single training example in each leaf and, by the previous proposition, its number of testing nodes is thus given by $N_t = n - 1$, where n is the number of training samples. No decision tree can thus have more than $n - 1$ test nodes, no matter how it is constructed.

[2]The difference comes from the fact that the randomly selected feature might be constant in the node and thus will lead to this node being pruned or a new feature to be selected, depending on the implementation. This situation is however unlikely when all features are numerical.

In regression, where each example has often a different output value, this upper bound is often reached with fully grown trees. In classification however, especially when the number of classes is low, tree growing is often stopped earlier because leaves contain only examples of the same class and thus can not be splitted further. A reasonable lower bound on tree complexity for a given learning sample can be obtained by measuring the complexity of a single CART tree (fully grown, without any randomization) fitted on this sample. One indeed expects that randomization will on average only increase tree size. In Section 4.2.2, we will compare these two bounds with the actual average tree size for different values of K on neuroimaging datasets.

Expected number of required trees

If we build an ensemble of T trees having on average N_t test nodes per tree, we will have considered all in all $T \times N_t$ pockets of K features. We can thus approximate the expected value of T required to have observed all features at least once by:

$$\mathbb{E}(T) = \frac{1}{N_t}\mathbb{E}(D), \tag{4.13}$$

where $\mathbb{E}(D)$ is computed by Eqn. (4.1) if $K = 1$, and by Eqn. (4.11) if $K > 1$. Note that since N_t can not be computed accurately,

4.2.2 Illustration on neuroimaging datasets

In this section, we analyse what this theory emphasizes practically for high dimension and few sample settings as met in typical neuroimaging datasets. We first study the number of draws necessary to observe each feature depending on K, with n and m values as observed in two real neuroimaging datasets. Then, we investigate the number of testing nodes found for different values of K and compare it with the two bounds discussed earlier. Finally, we conclude on the expected number of trees required to have seen all features at least once. Our study is performed on the CRC and OASIS datasets, introduced in Section 3.3 of Chapter 3.

Estimation of D

For $K = 1$, the ratio $\frac{\mathbb{E}(D)}{m}$ grows as $\ln(m)$ when m increases. $E(D)$ thus increases faster than m. Using the closed-form approximation (4.7), $\mathbb{E}(D)$ can be computed even for very large values of m. Table 4.1 provides the expected number of draws $\mathbb{E}(D)$ for the two neuroimaging datasets with $K = 1$. These numbers are huge in both cases, and higher than the number of features as expected. Given that these datasets are composed of only a few dozens of instances, we will see later than the individual tree complexity will not be large enough to reduce by more than one order of magnitude the number of trees required to have explored all features.

For $K = \sqrt{m}$, the expression of $\mathbb{E}(D)$ in Eqn. (4.11) is much more complex to analyse and furthermore its computation, which involves many factorial terms, becomes intractable for large values of m (i.e., $m > 25,000$). The red curve in Figure 4.1 illustrates the evolution of $\mathbb{E}(D)$, computed from Eqn. (4.11), for small values of m. The blue curve, on the other hand, has been estimated empirically. For each value of m, we drew $K = \sqrt{m}$ numbers at random among m repetitively until each number was drawn at least once. The y-axis value represents the number of draws D so obtained averaged over ten repeated experiments. This plot suggests that the tangent of the curve has now a slope decreasing with m and moreover, that the ratio $\frac{\mathbb{E}(D)}{m}$ is smaller than 1. Actually, we hypothesize that a reasonable approximation of $\mathbb{E}(D)$ for $K = \sqrt{m}$ can be obtained

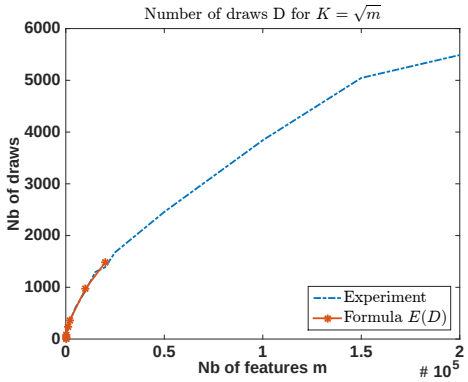


Figure 4.1 – Evolution of the experimental number of draws D necessary to have drawn each feature and of the theoretical expected value $\mathbb{E}(D)$ regarding the number of features m for $K = \sqrt{m}$.

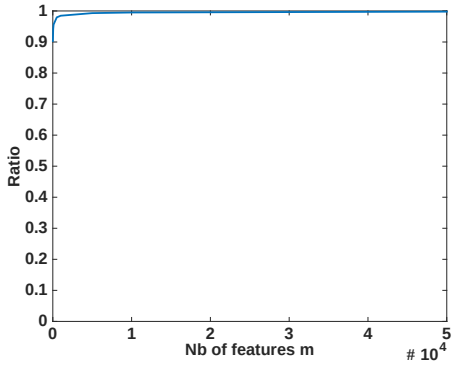


Figure 4.2 – Evolution of the ratio $\frac{\mathbb{E}(N_K)}{K}$ depending on the number of features m for $K = \sqrt{m}$.

with the following ratio:

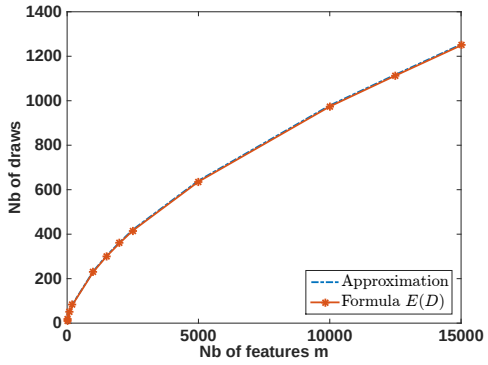
$$\mathbb{E}(D)_{approx} = \frac{\mathbb{E}(D)_1}{\sqrt{m}}, \tag{4.14}$$

where $\mathbb{E}(D)_1$ denotes the theoretical expected number of draws for $K = 1$. This approximation corresponds to sampling the K features *with* replacement instead of *without* replacement as used to derive Eqn. (4.11). This approximation thus overestimates the true $\mathbb{E}(D)$. It should however be a good approximation in particular for large m. Indeed, the probability that a given feature is selected among the K features when they are sampled with replacement is given by:

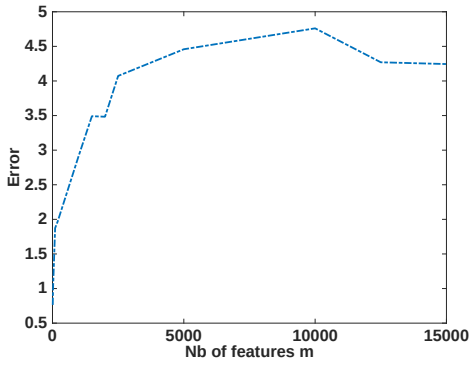
$$1 - \left(\frac{m-1}{m}\right)^K,$$

i.e., one minus the probability that this feature is not selected in the sample. Let us denote by $\mathbb{E}(N_K)$ the average number of distinct features among the K selected ones. $\mathbb{E}(N_K)$ can be computed by summing the previous probability over all features, which yields:

$$\mathbb{E}(N_K) = \sum_{i=1}^{m} \left(1 - \left(\frac{m-1}{m}\right)^K\right) = m\left(1 - \left(\frac{m-1}{m}\right)^K\right).$$



(a) Theoretical and approximative expected number of draws $\mathbb{E}(D)$ for $K = \sqrt{m}$.



(b) Error $(\mathbb{E}(D)_{approx} - \mathbb{E}(D)_{theor})$.

Figure 4.3 – Approximation of the expected number of draws $\mathbb{E}(D)$ for $K = \sqrt{m}$.

Table 4.1 – Estimation of $\mathbb{E}(D)$ and $k\sigma$ for CRC and OASIS datasets. n and m stand respectively for the number of samples and features in each dataset.

	$\mathbb{E}(D)$		$k\sigma$	n	m
	$K = 1$	$K = \sqrt{m}$	$K = 1$		
CRC	$2.83\ 10^6$	$6.02\ 10^3$	$1.26\ 10^6$	45	$219,727$
OASIS	$4.22\ 10^6$	$7.48\ 10^3$	$1.83\ 10^6$	100	$318,795$

As shown in Figure 4.2, the ratio $\frac{\mathbb{E}(N_K)}{K}$ when $K = \sqrt{m}$ converges towards 1 very rapidly as m grows. This means that on average, when $K = \sqrt{m}$ and m is large, the number of features covered when sampling with replacement is very close to the number of features covered when sampling without replacement and $\mathbb{E}(D)_{approx}$ is thus expected to be close to the true value of $\mathbb{E}(D)$ in such case. This is confirmed in Figure 4.3(a) that shows that $\mathbb{E}(D)_{theor}$ and $\mathbb{E}(D)_{approx}$ can not be distinguished. Figure 4.3(b) confirms that $\mathbb{E}(D)_{approx}$ overestimates the theoretical true value $\mathbb{E}(D)_{theor}$ and that the overestimation is negligible in comparison with the total number of draws necessary.

Table 4.1 provides the expected number of draws $\mathbb{E}(D)$ for the two neuroimaging datasets, using the approximation (4.14) for $K = \sqrt{m}$. We also provide standard deviation $k\sigma$ from Bienaymé-Chebyshev inequality (cf. Equation 4.10) with a confidence of 95% for $K = 1$. We need several millions of draws on average in order to have seen each feature at least once for $K = 1$ and several thousands of draws for $K = \sqrt{m}$. The standard deviation is about half of the number of draws. Therefore, the number of draws necessary in the case of $K = 1$ can be as large as 4 millions for the CRC data and 6 millions for the OASIS data, which is considerable.

Estimation of N_t

Table 4.2 shows the estimations of the tree complexity for both neuroimaging datasets. The first column shows the number of testing nodes included in a fully grown decision tree while the fourth column shows the number of testing nodes given by Proposition 4. The average number of testing nodes per trees estimated from a Random Forest of 100

Table 4.2 – Number of testing nodes in a single tree without randomization and in a Random Forests ensemble of 100 trees for the CRC and OASIS datasets. n and m stand respectively for the number of samples and features in each dataset.

	N_t			$n-1$	m	$\frac{m}{n}$
	1 tree	100 trees	100 trees			
		$K=1$	$K=\sqrt{m}$			
CRC	3	11.09	3.14	44	219727	4882.8
OASIS	6	23.67	7.06	99	318795	3187.9

Table 4.3 – Estimation of $\mathbb{E}(T)$ for CRC and OASIS datasets. n and m stand respectively for the number of samples and features in each dataset.

	$\mathbb{E}(D)$		$\mathbb{E}(T)=\frac{\mathbb{E}(D)}{N_t}$				n
			N_t for 1 tree		N_t for 100 trees		
	$K=1$	$K=\sqrt{m}$	$K=1$	$K=\sqrt{m}$	$K=1$	$K=\sqrt{m}$	
CRC	$2.83\ 10^6$	$6.02\ 10^3$	$9.43\ 10^5$	$2.01\ 10^3$	$2.55\ 10^5$	$1.92\ 10^3$	45
OASIS	$4.22\ 10^6$	$7.48\ 10^3$	$7.03\ 10^5$	$1.25\ 10^3$	$1.78\ 10^5$	$1.06\ 10^3$	100

trees is provided in columns 2 and 3 for $K=1$ and $K=\sqrt{m}$ respectively.

As expected, because CART trees do not involve any randomization, their number of testing nodes is smaller than that of Random Forests trees. With $K=1$, Random Forests trees are about four times larger than single CART trees, while, with $K=\sqrt{m}$, tree size is only marginally larger. In this latter case, the increase of tree complexity due to randomization is compensated by the reduction of complexity due to bootstrapping (which reduces by about 30% the effective size of the learning sample). On the other hand, the upper bound on tree complexity as given in Proposition 4 overestimates very strongly the actual tree sizes, by about a factor 4 when $K=1$ and about a factor 15 when $K=\sqrt{m}$.

Estimation of T

As we have estimated $\mathbb{E}(D)$ and N_t, we are now able to compute the expected number of trees $\mathbb{E}(T)$ required. Table 4.3 provides the theoretical $\mathbb{E}(T)$ for both neuroimaging datasets. We display in this table $\mathbb{E}(T)$ computed when N_t is given by one single tree or by the average on a forest of one hundred trees grown with the corresponding K setting. For $K=\sqrt{m}$, $\mathbb{E}(D)$ is given by the approximation suggested in Equation (4.14).

For $K=\sqrt{m}$, the minimum number of trees suggested by $\mathbb{E}(T)$ is around one thousand trees, depending on the database. This number is not too large and the more we have data instances, the lower we can expect it to be. When one feature is drawn at

each node ($K = 1$), observing each feature is a much slower process however, which requires at the minimum in the order of hundreds of thousands of trees in average.

Discussion

The purely combinatorial analysis performed in this section shows that many trees are required to have some minimum guarantee that each feature is seen at least once during the ensemble construction. If not enough trees are grown, then zero feature importance scores might originate from the corresponding features not having been seen at all during the tree construction, and not from real feature irrelevance or masking effects.

In practice, it is important to note that the numbers of trees, as reported in Table 4.3, are not expected to be sufficient to obtain reliable and stable feature importance scores. In principle, each feature has to be tested more than once and in different configurations for its importance score to be reliably estimated by the ensemble. The impact of K on the stability of estimated importance scores, for a given T, is not clear at this stage. When $K = 1$, each feature selected at a node will be used to split this node and this split will thus contribute to one term in the computation of the importance of this feature. On the other hand, when $K > 1$, even if K features are evaluated at each node, only one of them is eventually selected to split the node and thus each test node only contributes with one term to the importance of a single feature, exactly as when $K = 1$. Since $K > 1$ leads to smaller trees, it means that, for a fixed T, less impurity reduction terms will be computed when K grows, which could have an impact on stability.

To complement the theoretical analysis and analyse this effect, we will carry out experiments in the next section to study, empirically, the stability of importance scores as a function of the number of trees in the forest and of the setting of K.

4.3 Empirical study

Importance scores are obtained as averages over several trees. Given that trees are constructed independently of each other, as the number of trees T grows to infinity, feature importance scores converge towards their population values. We are interested in this section in assessing how many trees are required for importance scores to have reached convergence or the corresponding feature rankings to have reached some sort of stability. These numbers of trees will then be compared with the theoretical values computed in the previous section.

We first describe in Section 4.3.1 the experimental protocol and the stability measures that we propose. Then, experiments are carried out on artificial datasets in Section 4.3.2, to provide comparison baselines, and on real neuroimaging datasets in Section 4.3.3.

4.3.1 Stability measures and protocols

Stability of feature importance scores and rankings for a given number of trees T will be studied against two sources of randomization: randomization introduced during the tree construction (through bootstrap sampling and random feature selection at each node) and randomization due to the learning sample. To study the effect of the first randomization, we will fix the learning sample and construct Q forests of T_{max} trees with different random seeds. Subsets of T trees will then be drawn from each forest and the stability of the Q importance vectors so obtained will be assessed using the

different metrics detailed below. The same experiment but using a different learning sample for each forest of T_{max} trees will then be carried out to study the impact of learning sample randomization. In the case of artificial datasets, the different learning samples will be drawn from a large pool of randomly generated examples. In the case of the real datasets, the different learning samples will be obtained by randomly sampling, without replacement, 80% of the original dataset. Unless otherwise stated, T_{max} will be set respectively to 2500 and $100,000$ for the artificial and the real datasets and $Q = 10$ forests will be trained.

Stability of the resulting Q importance vectors will be assessed through several metrics. We will first show box-plots of importance scores over the Q repetitions for a selection of features. Following [Saeys et al., 2008], we will also measure the stability of the rankings derived from the importance scores by computing the average similarity between all pairs of rankings, with similarity measured by Pearson correlation, Spearman rank correlation and the Jaccard index of the top $x\%$ variables.

More formally, let $\mathbf{s}_i = (s_1^i, s_2^i, \ldots, s_m^i)^T$, with $i = 1, \ldots, Q$, denote the Q importance vectors for the m features. The stability of the resulting feature rankings will be computed as follows:

$$\text{Stab}(\mathbf{s}_1, \ldots, \mathbf{s}_Q) = \frac{2}{Q(Q-1)} \sum_{i=1}^{Q-1} \sum_{j=i+1}^{Q} S(\mathbf{s}_i, \mathbf{s}_j),$$

with $S(\mathbf{s}_i, \mathbf{s}_j)$ a measure of the similarity between $\mathbf{s}_i$ and $\mathbf{s}_j$. The similarity as measured through the Pearson correlation coefficient is defined as follows:

$$S(\mathbf{s}_i, \mathbf{s}_j) = \frac{cov(\mathbf{s}_i, \mathbf{s}_j)}{\sigma_{\mathbf{s}_i} \sigma_{\mathbf{s}_j}}, \tag{4.15}$$

where cov is the covariance and σ the standard deviation. Denoting by $\text{rk}(s_l^i)$ the rank[3] of the importance s_l^i of feature l in the vector $\mathbf{s}_i$, the similarity measured through the Spearman rank correlation coefficient is defined as:

$$S(\mathbf{s}_i, \mathbf{s}_j) = 1 - \sum_{l=1}^{m} \frac{(\mathbf{rk}(s_l^i) - \mathbf{rk}(s_l^j))^2}{m(m^2-1)}. \tag{4.16}$$

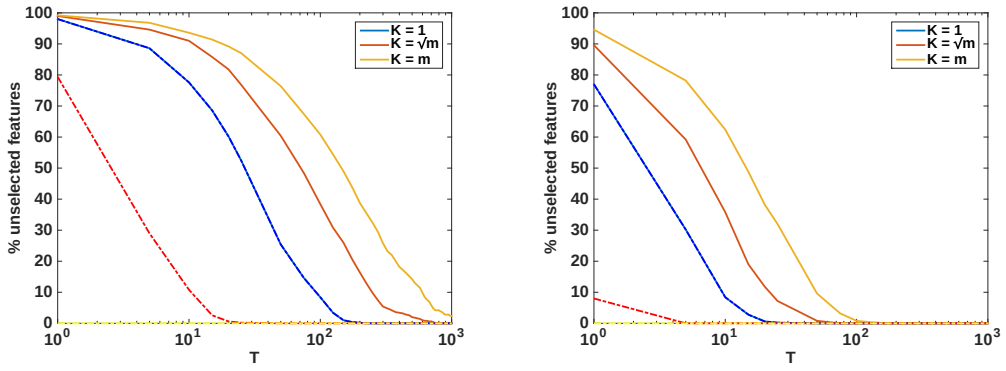
Finally, Jaccard index similarity is defined as:

$$S(\mathbf{s}_i, \mathbf{s}_j) = \frac{|\ \mathbf{f}^{x\%}(\mathbf{s}_i) \cap \mathbf{f}^{x\%}(\mathbf{s}_j)\ |}{|\ \mathbf{f}^{x\%}(\mathbf{s}_i) \cup \mathbf{f}^{x\%}(\mathbf{s}_j)\ |}, \tag{4.17}$$

where $\mathbf{f}^{x\%}(\mathbf{s}_i)$ is the subset of features included in the $x\%$ top ranked variables according to the importance scores in $\mathbf{s}_i$. All three similarity measures are such that perfect stability corresponds to $\text{Stab}(\mathbf{s}_1, \ldots, \mathbf{s}_Q) = 1$. In the case of the two correlation measures, stability lies in $[-1, 1]$, while in the case of the Jaccard index, stability lies in $[0, 1]$, with 0 meaning that all rankings put different features in their top.

As a last measure of stability in the case of artificial datasets, we will also compute the average rank of the truly relevant features over the Q rankings. If these features are all at the top of the rankings then, this average rank should be equal to its smallest possible value, i.e., $(R+1)/2$, with R the number of truly relevant features. This value is thus a way to monitor the quality of the ranking, in addition to its stability. Unlike previous metrics, the average rank is insensitive to the exact relative ranking of the relevant features.

[3] An average rank is attributed to features with similar importance scores.



(a) Artificial data (50 samples). (b) Artificial data (500 samples).

Figure 4.4 – Evolution of the percentages of unselected (plain lines) and unseen (dotted lines) features with T for different values of K on the artificial datasets: left, for a learning sample of size 50, right, for a learning sample of size 500. The x-axis is in log scale. Note that for $K = 1$, the percentages of unselected and unseen features are equal.

4.3.2 Artificial dataset

We first study in this section the stability of importance scores in the case of an artificial dataset. The objective of these experiments is to gain some intuitions about the impact of data and method parameters (m,n, K, and T) on the stability of importance scores in a controlled setting. This intuition will then be used as a basis for the analysis of the results on real neuroimaging datasets in Section 4.3.3.

Data generation

Let us denote by $LS = (X, Y)$ the learning set where $X = (X_1, \ldots, X_m) \in \mathbb{R}^{n \times m}$ and $Y \in \{0, 1\}^n$. All m input variables are independent and normally distributed, i.e., $X_i \sim \mathcal{N}(0, 1), \forall i = 1, \ldots, m$. The output vector Y only depends on the first $R = 5$ features through the following function:

$$Y = sgn\left(\sum_{i=1}^{R} w_i X_i\right), \tag{4.18}$$

where the values of w_i are chosen such that $w_1 = 1$, $w_2 = 0.9$, $w_3 = 0.8$, $w_4 = 0.7$ and $w_5 = 0.6$. All remaining $m - R$ features are irrelevant. Because of the decreasing weights and all features being equally distributed, the first five features should in principle receive also decreasing importance scores. To make the problem harder to solve, 1% of the output values are randomly flipped.

To study the impact of the number of features, we generated two datasets respectively with 500 and 1000 features, among which $R = 5$ are relevant and respectively 495 and 995 are thus irrelevant. We generated a total of 500 instances but study later also smaller learning samples of 50 instances.

Unseen versus unselected variables

Figure 4.4 shows the evolution with T of the percentage of features that are respectively unseen and unselected during the tree construction, as a function of K (among

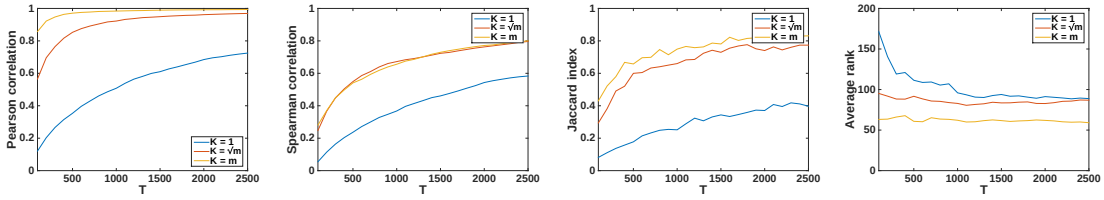
$\{1, \sqrt{m}, m\}$), in Figure 4.4(a) for a learning sample of size 50 and in Figure 4.4(b) for a learning sample of size 500. The point where the percentage of unseen features reaches 0 corresponds to one observation of the number of trees required to have seen each feature at least once as studied in the previous section. On this problem, which is better conditioned than the neuroimaging datasets, this number is on the order 250, 50, and 1 respectively for $K = 1$, $K = \sqrt{m}$, and $K = m$, with 50 learning examples. These numbers are reduced by about one order of magnitude when going from 50 to 500 examples. It is interesting to compare these numbers with the number of trees requires to have selected each feature at least once for splitting a node (i.e., when the percentage of unselected features reaches 0). When $K = 1$, these two numbers perfectly match since as soon as a feature is evaluated, it is selected to split. When $K > 1$ on the other hand between one (for $K = \sqrt{m}$) and two (for $K = m$) orders of magnitude more trees are needed to have selected all features with respect to having seen all of them. As expected, more trees are required when $K = m$, because in this case, the only source of randomization is bootstrap sampling and masking effects are therefore more present. Nevertheless, even for this setting, all features end up being selected at least once in the forest. Note that in the ideal case, only the five relevant features should be selected, as all other features are irrelevant by construction. The fact that all features end up being selected is due to the finite size of the learning sample, which leads irrelevant features to receive non zero impurity reduction scores in particular at the deeper nodes in the trees.

Stability of importance scores

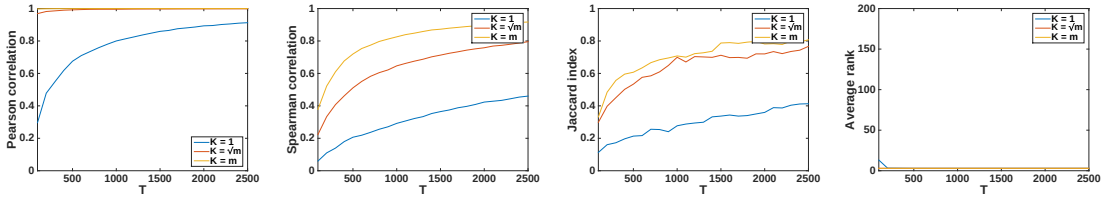
Figure 4.5 illustrates the effect of different parameter settings on the stability of importance scores measured by the four metrics explained earlier: Pearson correlation, Spearman correlation, Jaccard index (with $x = 5\%$) and the average rank of the five relevant features. These plots have been obtained for a fixed learning sample.

As expected, whatever the metric, stability is always monotonically increasing with the size T of the forest. When $K > 1$, Pearson correlation reaches its maximum value for large values of T, while Spearman correlation reaches much smaller values. This is a consequence of the fact that rankings are expected to be more unstable than absolute importance scores, as a small change of importance scores can lead to important changes in the ranking. Average Jaccard index for larger K values can be as high as 0.8, which means that in average two rankings have in common 22 features among their top 25 features. Comparing Figure 4.5(a) with Figure 4.5(b) (and Figure 4.5(c) with Figure 4.5(d)) shows that increasing the number of samples n from 50 to 500 significantly improves the stability of importance scores as measured with Pearson correlation. However, such improvement is not so clear regarding the measures of stability based on ranking (Spearman correlation and Jaccard index), except for $K = m$ and Spearman correlation. A significant improvement is however observed in the average rank for the five relevant features (in the rightmost plot of each row in Figure 4.5). For the smallest sample size, the average rank is very high suggesting that some truly relevant features can not be distinguished from irrelevant ones using importance scores. For the largest sample size however, the optimal value of 3, corresponding to all 5 relevant features at the top of the ranking, is reached by all methods. Comparing Figure 4.5(d) (resp. 4.5(c)) with Figure 4.5(a) (resp. 4.5(b)) shows that doubling the number of irrelevant features only moderatly decreases the three stability metrics. The strongest impact is observed on the average rank for the smallest learning sample size.

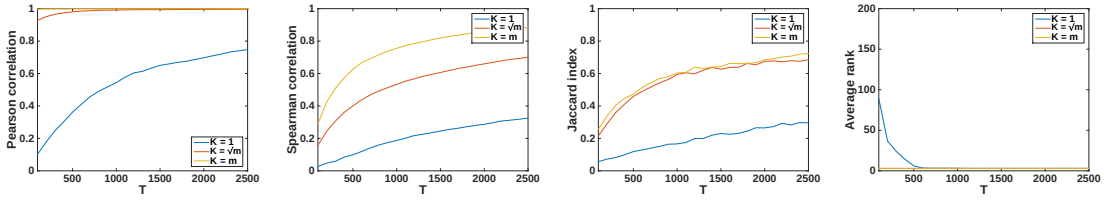
Figures 4.6 and 4.7 show box plots of importance scores, respectively for 50 and 500 samples, for a selection of 10 features: the five relevant features (from 1 to 5) and the five irrelevant features of lowest average rank among all irrelevant features at $T = 1000$



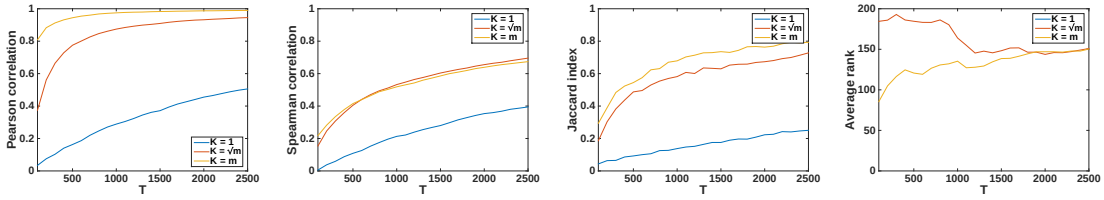
(a) Artificial data (50 samples and 500 variables).



(b) Artificial data (500 samples and 500 variables).



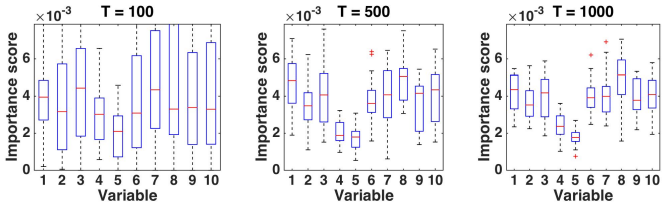
(c) Artificial data (500 samples and 1000 variables).



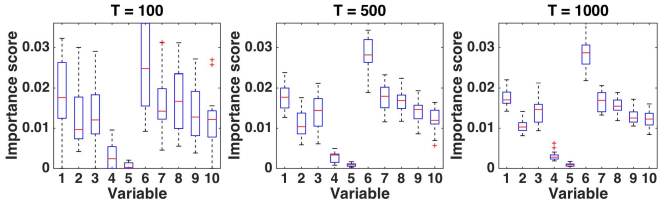
(d) Artificial data (50 samples and 1000 variables).

Figure 4.5 – Artificial data. Evolution with T of the Pearson correlation, Spearman correlation, and Jaccard index (5%) stability and of the average rank of the five relevant features.

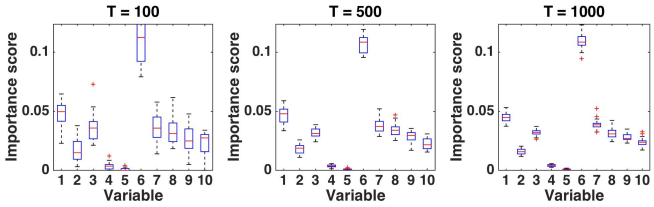
(from 6 to 10). It is worth noting that the y-axis is not at the same scale for the different values of K. Indeed, the sum of importance scores is stable (it is equal to the total variance of the output [Louppe et al., 2013]) and, when K increases, the sum is distributed among less features leading to an overall increase of the importance scores of the selected features. With the smallest sample size, Figure 4.6 shows that, whatever the value of K, some irrelevant features have a higher importance score than some of the relevant features. On the contrary, in Figure 4.7, the relevant features are the features of highest importances for all values of K. These observations are consistent with the average rank plots in Figure 4.5. These figures also illustrate that the stability, as measured here by the variance of importance scores, decreases with an increase of T and K. In particular, a high K value requires less trees to have stable importance scores. This is expected as a high value of K corresponds to less randomization in the tree construction. Figure 4.8 shows the box plots for both sample sizes and $K = \sqrt{m}$ when 500 more irrelevant features are added to the dataset. When comparing with box



(a) $K = 1$.



(b) $K = \sqrt{m}$.

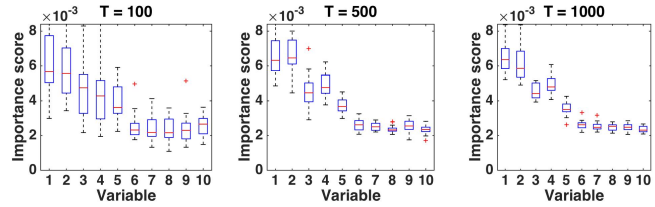


(c) $K = m$.

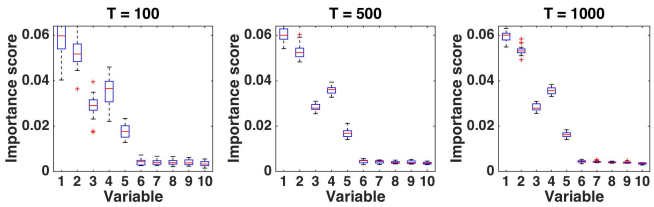
Figure 4.6 – Artificial data (50 samples and 500 variables). Box plots for the importance scores computed for 20 runs of RF algorithm. The first five features correspond to the relevant features while the next five features correspond to the five irrelevant features with the lowest average rank for $T = 1000$.

plots in Figures 4.6 and 4.7, we observe a slight decrease of importance scores, which is due again to the fact that the sum of importances is fixed and distributed among more features. Whatever the sample size, we do not observe however a significant increase of importance score variances. This is again consistent with what we observed in Figure 4.5.

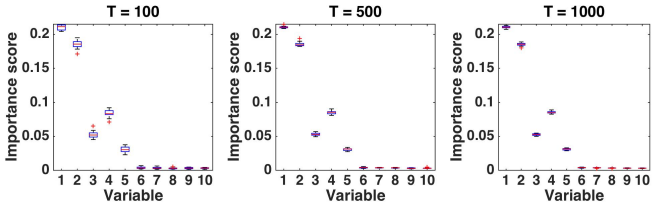
In previous experiments, the learning sample was fixed and variations of importance scores were only caused by the randomization introduced in the forest construction. To evaluate the additional variance caused by the learning sample randomization, we repeated the same experiments using this time a different sub-sample of size 50 to grow each of the ten forests. The results of these experiments are shown in Figure 4.9. Figure 4.9(a) is exactly the same as Figure 4.5(a) and is reproduced here only to ease comparisons. A modification of the learning set drastically decreases the values of the three stability measures. This is expected given the nature of the problem. When the learning sample is changed, the "best" irrelevant features, and therefore their rankings according to importance scores, are expected to change completely and therefore even when T and K are large, rankings and importance scores for these features are highly unstable. Only the relevant features can be common between the rankings but they represent only 1% of the total number of features. It is interesting to note that when the learning sample is changed, increasing K does not seem anymore to have a positive impact on stability. Both settings of K can not be distinguished anymore. This can be explained. While increasing K improves stability for a fixed learning sample, it increases the de-



(a) $K = 1$.

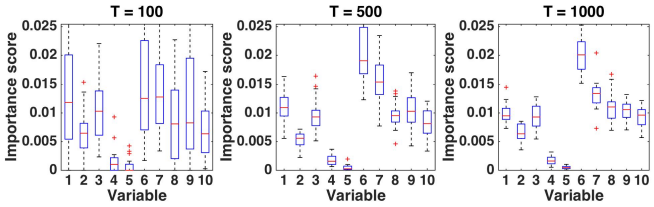


(b) $K = \sqrt{m}$.

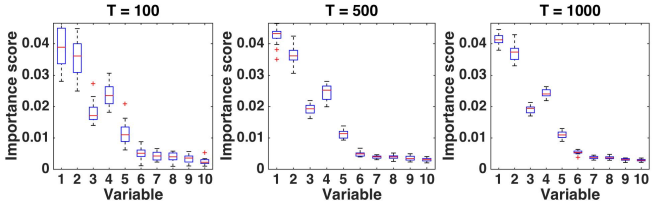


(c) $K = m$.

Figure 4.7 – Artificial data (500 samples and 500 variables). Box plots for the importance scores computed for 20 runs of RF algorithm. The first five features correspond to the relevant features while the next five features correspond to the five irrelevant features with the lowest average rank for $T = 1000$.

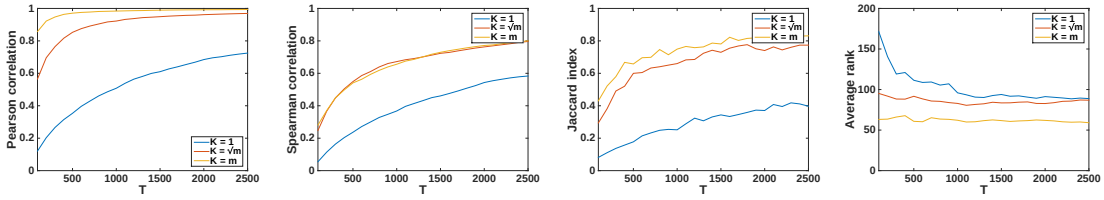


(a) 50 samples, $K = \sqrt{m}$.

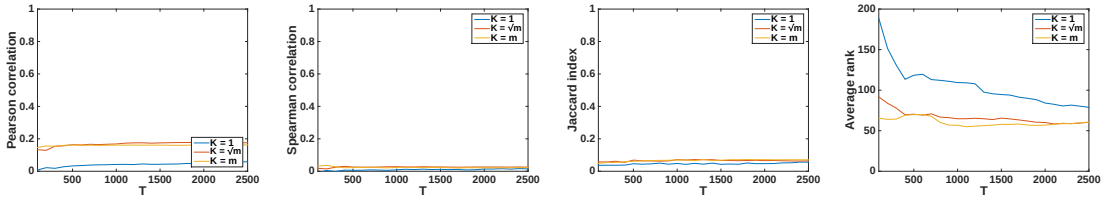


(b) 500 samples, $K = \sqrt{m}$.

Figure 4.8 – Artificial data (1000 variables). Box plots for the importance scores computed for 20 runs of RF algorithm. The first five features correspond to the relevant features while the next five features correspond to the five irrelevant features with the lowest average rank for $T = 1000$.



(a) Artificial data (50 samples and 500 variables) with a fixed learning sample.



(b) Artificial data (50 samples and 500 variables) with different learning samples.

Figure 4.9 – Artificial data. Impact of learning sample variability.

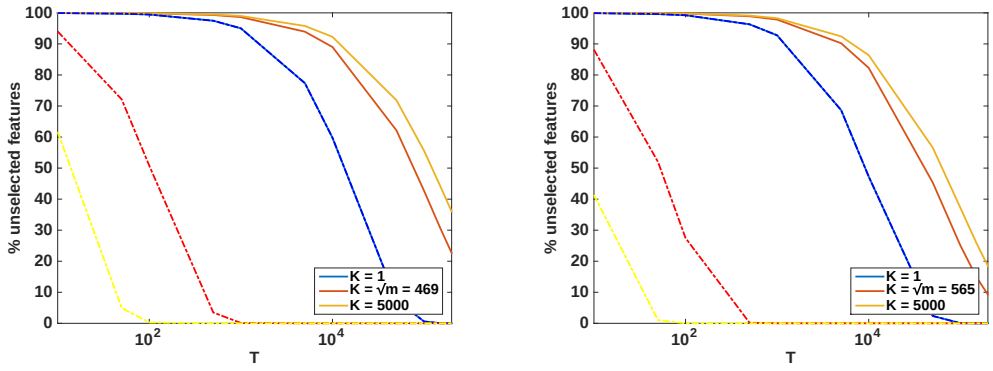
pendence of the model on the learning sample, which thus reduces stability when the learning sample is changed. In addition, while $K = m$ leads to a better average rank of the relevant features than $K = \sqrt{m}$ on a single learning sample, this is not the case anymore when the learning sample is changed. Both settings are now equally good in terms of average rank, while $K = \sqrt{m}$ leads to improved computing times.

4.3.3 Neuroimaging datasets

We now carry out similar experiments on two real datasets. They are the CRC and the OASIS datasets, already exploited in Section 4.2.2. They both include more than $200,000$ features and less than 100 samples. Note that average rank plots can not be shown on these datasets as the truly relevant features are unknown.

Unseen versus unselected variables

Figure 4.10 displays the percentages of unseen and unselected features as a function of T for different values of K (1, $\sqrt{m} \simeq 500$, and 5000). For these tests, we built a forest of $T = 200,000$ randomized trees and we counted the number of features unseen or not appearing among the testing attributes for subsets of this forest of increasing sizes. For $K = 1$, about $10,000$ trees are necessary to have 50% percent of the attributes either evaluated once or used as testing attributes. All features have been used as testing attributes for a forest of in the order of $100,000$ trees, which corresponds with the numbers provided in Table 4.3. When $K > 1$, having observed a feature does not mean that it will necessarily be used as testing attribute, as the algorithm then embeds a feature selection procedure. The percentage of unselected features thus decreases more slowly for $K = \sqrt{m}$ and $K = 5,000$ than for $K = 1$. With $K = \sqrt{m}$, the percentage of unselected features actually has not reach zero at $T = 200,000$. At this value, 30% (resp. 10%) of the features do not appear at any node in the forest on the CRC (resp. OASIS) dataset. On the other hand, on both datasets, the percentage of unseen features reaches zero for $K = \sqrt{m}$ with in the order of $1,000$ trees, which is again consistent with the results in Table 4.3. At this value of T, the percentage of selected features is very low on both datasets. The slow decrease of the percentage of unselected features with T when $K > 1$ is also a consequence on these datasets of the spatial correlations that



(a) CRC data. (b) OASIS data.

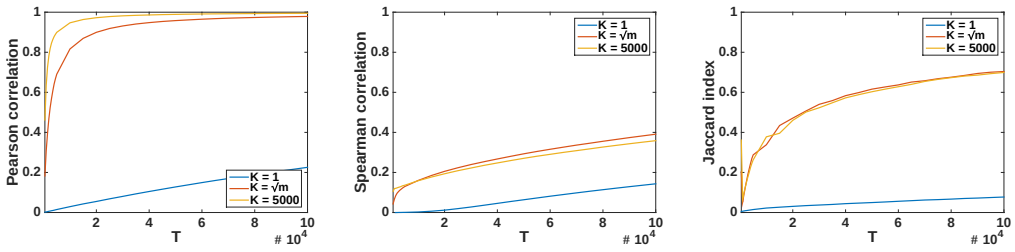
Figure 4.10 – Evolution of the percentages of unselected (plain lines) and unseen (dotted lines) features with T for different values of K on the CRC (left) and OASIS (right) datasets. The x-axis is in log scale. Note that for $K = 1$, the percentages of unselected and unseen features are equal.

exist between the features, which makes them redundant and thus reinforces masking effects. Note that, as for the artificial dataset, one does not expect that all features should be selected in the forest. Ideally, only the relevant ones should be selected. Although their numbers is unknown, one does not expect that all features are relevant on these datasets. Despite this, plots in Figure 4.10 suggests that all features will be eventually selected in the forest even for the larger K value. Again, this is due to the finite size of the learning sample and the randomization, which gives a chance even to irrelevant features to be selected, especially at deeper nodes in the trees.

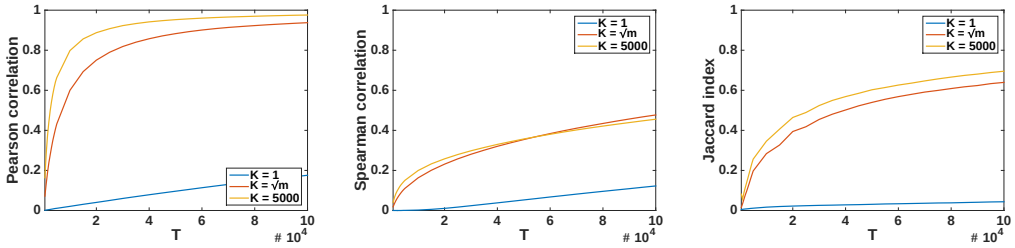
Stability of importance scores

Figure 4.11 illustrates the evolution of the three stability measures depending on T. For $K > 1$, the Pearson correlation stability converges towards 1 after about $100,000$ trees for both datasets. The convergence is slightly slower on the OASIS dataset that shows a higher value of m. The measures for the rank stability converge much slower than the Pearson correlation. Spearman correlation stability remains below 0.5 even with $100,000$ trees. Jaccard index similarity (with $x = 1\%$) eventually reaches about 0.7 on both datasets as soon as $K > 1$. On the CRC dataset, this means that on average two rankings have in common 1809 features out of 2197 on the CRC dataset and 2625 features out of 3188 on the OASIS dataset. Using $K = 1$ leads to very unstable rankings, while increasing K from about 500 to 5000 have only a limited impact on stability, except when it is measured by Pearson correlation.

Figures 4.12 and 4.13 show box plots for a few selected features. Given that the truly relevant features are unknown on these datasets, the ten first features shown in these box plots were selected as follows (for each K value separately): we computed the average importance of the features over the ten forests of size $100,000$ and selected the ten features of highest average importance. When $T = 1000$ and $T = 10,000$, the two features of highest average importance can be added to the box plots (at position 11 or 12) if these features are different from the ten first features displayed. When such features are added, then they are also displayed on the box plot corresponding to $T = 100,000$ at positions 11 to 14. For instance, in Figure 4.12(a), the eleventh and twelfth features are the features of highest importance for $T = 1000$, which have lower importance than



(a) CRC dataset.



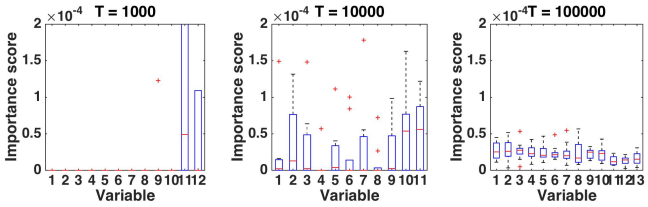
(b) OASIS dataset.

Figure 4.11 – Real datasets. Evolution of the Pearson correlation, Spearman correlation, and Jaccard index (1%) stability as a function of the number of trees T.

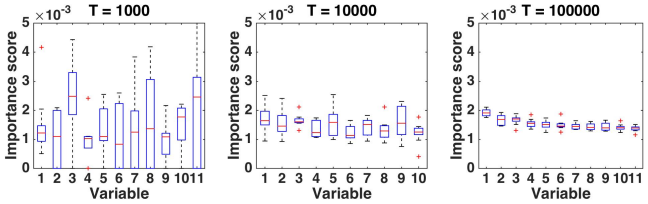
the ten best ones for $T = 100,000$. Another feature shows a higher median importance than the ten best ones for $T = 10,000$ as represented. This feature appears then at the thirteenth position for $T = 100,000$, which confirms that it has a lower importance than the first ten features for this forest size.

The very high variance of importance scores is very clear from these box plots, on both datasets. Only when $T = 100,000$ and for the larger values of K, the relative ranking between the ten top ranked features seems to have reached stability. Note that even in this latter case, the ranking being stable does not mean that all features at its top are truly relevant features. As clearly shown for example in Figure 4.6 on the artificial dataset, importance scores of irrelevant features can be greater than importance scores of truly relevant ones and still have a low variance at fixed LS. From these box plots, and also the stability curves in Figure 4.11, it also appears that the number of trees predicted by the theoretical analysis in Section 4.2.2 is clearly not sufficient to obtain stable importance scores. For $K = \sqrt{m}$, on the order of 10^3 trees are necessary to have seen all features, but with this number of trees, stability is very low and importance scores have a very high variance. At least one or two orders of magnitude more trees are required to reach stability.

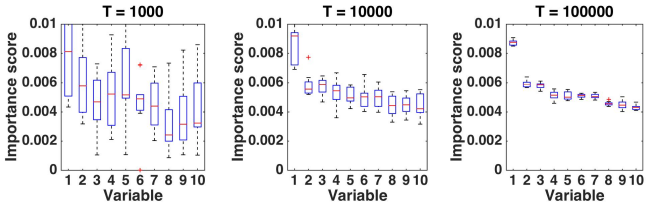
On real datasets, it is not possible to generate several learning samples to study the impact of learning sample variability on stability. Following [Saeys et al., 2008], we nevertheless reproduced the experiment of Figure 4.11, with the difference that the ten forests are now grown each from a different learning sample, obtained by sampling without replacement 80% of the original dataset. The resulting stability curves are shown in Figure 4.14 and 4.15, respectively for the CRC and the OASIS dataset (where Figures 4.11(a) and 4.11(b) have been reproduced to ease comparisons). As in the case of the artificial data, this extra-randomization reduces very much stability. For example, Jaccard index stability does not exceed now 0.2, which means that only about



(a) $K = 1$.



(b) $K = \sqrt{m}$.



(c) $K = 5000$.

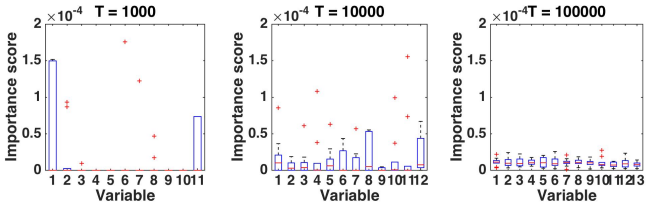
Figure 4.12 – CRC dataset. Box plots for the importance scores computed for 10 runs of RF algorithm. The first ten features are the features of highest average importance for $T = 100,000$. The first two features for $T = 1000$ and $T = 10,000$ are also displayed if different of the first ten (see the text for more details).

one third of the top features are shared between the rankings. Given that there is still an important overlap between the learning samples, the stability in response to true learning sample variability can be expected to be even lower. Interestingly, as for the artificial data again, increasing K with respect to the default setting has now a negative impact on stability, again probably because of a higher variance due to overfitting.

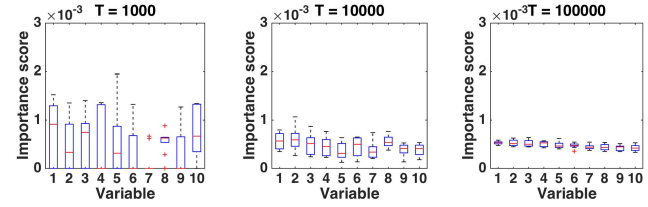
4.4 Discussion

In this chapter, we took an interest in the reliability of the importance scores provided by Random Forests. For high dimension and small sample size problems, null importance scores might be attributed to truly informative features only because they have not been seen during the whole learning process. This can have serious consequences for feature selection based on these scores. For instance, if only features with non zero score are selected for the learning stage, a lot of useful features could be wrongly dismissed and the classifier could suffer from bad performance.

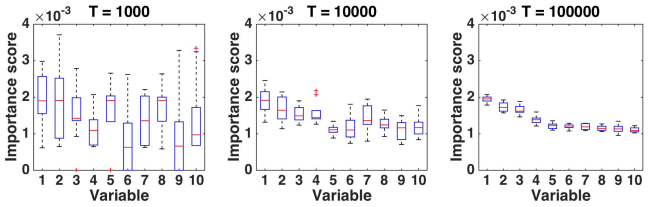
In the first part of the chapter, we investigated theoretically the expected number of trees required on average in order to have observed each feature at least once. This problem has been handled by using the theory of the *coupon collector's problem*. We evaluated the expected number of trees for two real datasets. We observed that increasing the parameter K from 1 to $\sqrt{m}$ allows to decrease significantly the expected number of required trees, going from the order of 10^5 trees to the order of 10^3. Although



(a) $K = 1$.



(b) $K = \sqrt{m}$.



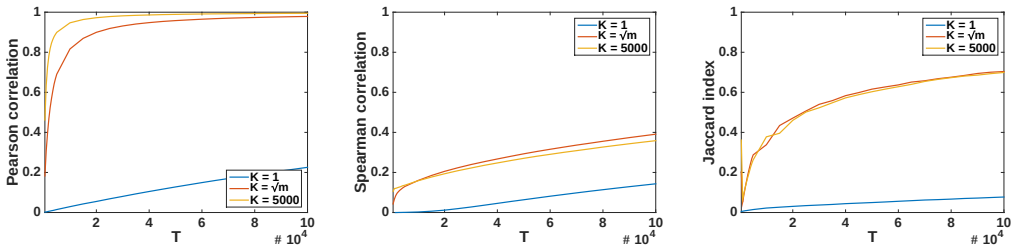
(c) $K = 5000$.

Figure 4.13 – OASIS dataset. Box plots for the importance scores computed for 10 runs of RF algorithm. The first ten features are the features of highest average importance for $T = 100,000$. The first two features for $T = 1000$ and $T = 10,000$ are also displayed if different of the first ten (see the text for more details).

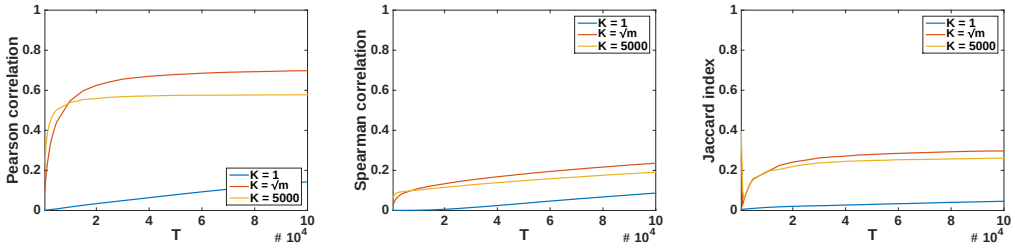
this number provides a useful information about the problem, it only constitutes a very minimal lower bound of the number of trees that are really required, as each feature should ideally be observed multiple times and in multiple configurations to yield reliable importance scores.

In the second part of the chapter, we therefore studied empirically the stability of importance scores for artificial and real datasets. On the artificial dataset, we used three different stability measures and the average rank of the relevant features to evaluate the variance of the importance scores depending on T, K, m and n caused by the randomization of both the Random Forests algorithm and the learning sample generation. We observed that stability is getting worse if the value of n or K is decreased or if the value of m is increased, with all other parameters remaining the same. Moreover, by observing the average rank evolution, we saw that a too low n value cannot lead to a detection of all relevant variables. Finally, learning sample variability leads to a very important decrease of stability. It also shows that although $K = m$ leads to more stable rankings at fixed learning sample, it actually performs equally to $K = \sqrt{m}$ when learning sample variability is taken into account. On the real datasets, we observed that, even with a high K value ($K = \sqrt{m}$ or $K = m$), about $100,000$ trees are necessary for stabilising the importance scores. It is worth noting that this value is one hundred times higher than the one advised by the theoretical analysis performed at the beginning of the chapter.

In conclusion, tree based variable importance scores are expected to be highly un-

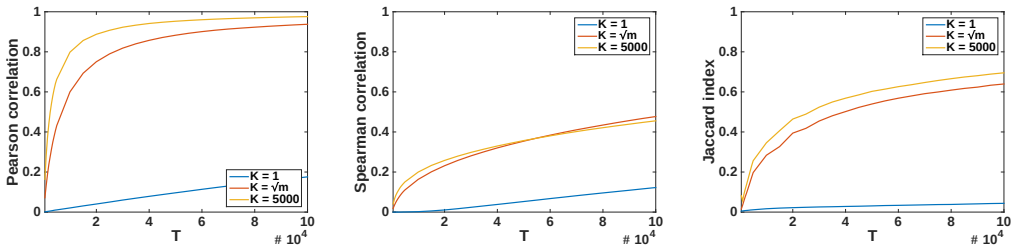


(a) Full sample size.

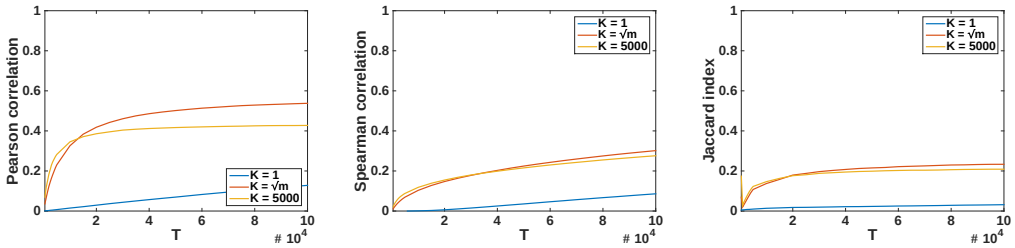


(b) Various subsamples of size $0.8\ n$.

Figure 4.14 – CRC dataset. Impact of learning sample variability.



(a) Full sample size.



(b) Various subsamples of size $0.8\ n$.

Figure 4.15 – OASIS dataset. Impact of learning sample variability.

stable in the small sample and high dimensional settings of neuroimaging datasets. In such situation, small tree size and high dimensionality makes it very hard for a given specific relevant feature to emerge as important. Increasing K improves stability at fixed LS but it increases overfitting and thus variance, which makes the resulting feature rankings not necessarily better than when using smaller K, at the expense furthermore of computing times. Increasing T has only a positive impact on stability, by getting rid of the randomization due to tree construction. Even if the number of trees is large however, obtaining very stable importance scores does not necessarily imply that

the most important variables are truly relevant, as clearly shown by the experiment with small sample sizes on the artificial datasets.

In the next chapters, we will develop approaches to circumvent some of these issues by taking into account specificities of neuroimaging datasets. Indeed, these datasets are such that features, corresponding to voxels, are spatially organized. We will develop several techniques in Chapter 5 to improve importance scores under the hypothesis that features that are close spatially should have similar importance scores. Under this hypothesis, information can then be shared between features that should improve stability of the resulting importance scores and thus reduce the needs in terms of the number of trees. We will also explore in this chapter and the next, techniques to highlight relevant groups of contiguous features, instead of isolated features, that therefore implicitly reduce the dimensionality and make the problem easier to solve. Finally, to confirm or not the relevance of the most important features found for a given ensemble size, we will develop permutation schemes at the group level in Chapters 6 and 7 to assess how much the position of each group in the importance ranking is due to instability or true relevance.

Chapter 5

Exploiting spatial and group structure in variable importance scores

Chapter overview

As explored in the previous chapter, the interpretation of importance scores for a typical neuroimaging problem can only be reliable if it is based on a forest composed of a very large number of trees. In this chapter, we thus study several methods to improve tree based importance scores for a fixed forest size. The proposed methods exploit either the intrinsic group structure or the existing correlations between features existing in neuroimaging data. After having introduced the problem, the datasets and the baseline results in Sections 5.1, 5.2 and 5.3 respectively, the main idea of Section 5.4 is to preprocess input data, compute importance scores on new input features and then infer importance scores for the initial features from the importance scores of the new features. In Section 5.5, we directly modify the Random Forests algorithm itself in order to compute importance scores. Methods in Section 5.6 modify importance scores a posteriori by applying them a postprocessing transformation. Each section contains experiments on artificial and (pseudo-)real datasets. We end this chapter by a short summary of our findings.

5.1 Introduction

In the previous chapter, we showed that very large forests need to be constructed in small sample size/high dimensional settings if one wants to have some minimal guarantees in terms of coverage of the features and stability of the importance scores. Even with very large ensembles, small sample size might make it very difficult for a given relevant feature to emerge in such setting due to severe overfitting. In this chapter, we investigate several techniques that aim at improving tree based variable importance scores, in terms of ranking quality and stability for a given forest size, by taking into account some specificities of neuroimaging problems. Our hope with these methods is to reduce the requirements in terms of number of trees with respect to standard importance scores, in particular in small sample size setting.

The idea to obtain such improvement is to exploit the structure that exists between features in neuroimaging datasets. In brain imaging, features indeed correspond to

voxels of 3D images and these voxels are therefore spatially organized. Two neighbouring voxels are expected to represent highly correlated information due to the spatial resolution. Given these correlations, one can reasonably assume that features that are spatially close to each other should receive similar importance scores. This assumption can then be leveraged to share information between features, which will implicitly reduce the dimensionality of the data. For functional images, we also expect features to be organized in regions, i.e. groups of spatially contiguous voxels, corresponding to different activities. Neuroscientists are more often interested in highlighting brain regions than isolated voxels in their statistical analyses. These regions are usually predefined through anatomical atlases that segment the brain a priori into the main activity regions of general interest. Again, the constraint that features in a group can receive similar importances or that only groups need to be properly ranked can be exploited to implicitly reduce the dimensionality of the data.

In this chapter, we explore several simple methods that modify the way tree based importance scores are computed so as to exploit either the spatial organization of the features or a pre-defined division of the features into groups. These methods are divided into three main families depending on how they modify the original importance scores: *preprocessing* methods modify the training data, *embedded* methods modify the training algorithm, and *postprocessing* methods modify feature importance scores a posteriori. The potential of these methods is illustrated on randomly generated artificial datasets and on one (pseudo-)real dataset.

We first describe the datasets and performance measures in Section 5.2. We then provide baseline results with standard Random Forests importance scores in Section 5.3, before going through preprocessing, embedded, and postprocessing methods respectively in Sections 5.4, 5.5, and 5.6.

5.2 Datasets and performance measures

In order to explore alternatives to standard tree-based importance scores, we propose to use artificial data in which relevant and irrelevant variables are known. These datasets are constructed in such a way that the truly informative features are organised in groups of correlated variables. Indeed, in neuroimaging analyses, we expect that relevant voxels can be grouped into spatially localized groups of voxels rather than in sparsely distributed patterns. We generated two artificial datasets. The first one is fully artificial, while the second one is derived from a real neuroimaging dataset.

Artificial datasets

We first generate artificial datasets corresponding to linear classification problems. Artificial datasets construction is inspired from the linear datasets construction used in [Huynh-Thu et al., 2012]. It is however adapted to enforce a group structure between the features. Figure 5.1 illustrates the procedure of dataset construction explained here below.

Each dataset contains m features denoted $(x_1, \ldots, x_m)$ that are divided a priori into g groups denoted $(G_1, G_2, \ldots, G_g)$, with the size of group G_i denoted $\#G_i$ (we use $m = 2000$ and $g = 50$ in all our experiments). We assume that features are ordered following the group distribution such that group G_i is composed of features $x_{(\sum_{k=1}^{i-1} \#G_k)+1}$ to $x_{\sum_{k=1}^{i} \#G_k}$, $\forall i = 1, \ldots, g$.

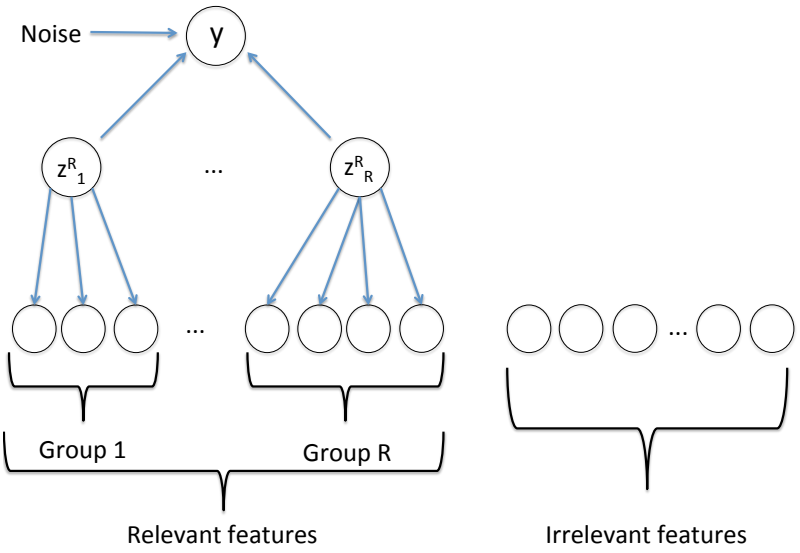


Figure 5.1 – Construction of artificial datasets.

To generate groups of random sizes, we proceed as follows. We draw $g-1$ cut-off values at random without replacement from $\{1,\ldots,m\}$. Denoting by $(c_1,\ldots,c_{g-1})$ these values in increasing order and defining $c_0=0$ and $c_g=m$, the size of the ith group ($i=1,\ldots,g$) is then set to c_i-c_{i-1}.

Among these groups, the R first ones are chosen as relevant and the remaining $I=g-R$ are chosen as irrelevant. Let us denote by G^R and G^I respectively the sets of relevant and irrelevant groups.

Values of the features in the irrelevant groups are drawn independently of each other from a normal distribution, i.e., $x_i \sim \mathcal{N}(0,1), \forall x_i \in g$ and $\forall g \in G^I$.

For each relevant group $G_k \in G^R$, one hidden variable z_k^R is first drawn from a normal distribution such that $z_k^R \sim \mathcal{N}(0,1)$ for $k=1,\ldots,R$. The output y is then computed from the z_k^R hidden variables as follows:

$$y = \text{sgn}\left(\sum_{k=1}^{R} w_k z_k^R\right),$$

where the values of the coefficients w_k are drawn uniformly in $[0,1]$. The hidden variables z_k^R are not put directly in the dataset. Instead, all the features within each relevant group are generated each as noisy copies of z_k^R, obtained by adding a normal $\mathcal{N}(0,1)$ noise to z_k^R. The motivation for this procedure is to create a non perfect correlation between the features within the relevant group, so that they are jointly more informative about the output than individually.

Finally, 1% of the output values have been randomly flipped to make the problem harder to solve.

Real dataset

The aim of the previous dataset is to provide a comparison baseline. The group structure and spatial correlation between features are simplistic and not expected to be realistic. In particular, they are unidimensional in the sense that groups are composed of contiguous features when they are ordered according to their dataset indices, i.e., a single dimension.

To provide a more realistic benchmark, we propose to derive a second pseudo-artificial problem from a real neuroimaging dataset, the CRC one (see Section 3.3). In such real dataset, truly relevant and irrelevant features are obviously unknown. In order to nevertheless permit a quantitative evaluation of feature rankings provided by the methods proposed in this chapter, we will consider below as truly relevant regions four regions from the AAL atlas (see [Tzourio-Mazoyer et al., 2002] and Appendix C) that have been previously highlighted in the literature on the classification between MCI converters and MCI stable patients on the basis of FDG-PET scans. More precisely, these four regions are the inferior parietal (right hemisphere), the middle temporal gyrus (right and left hemispheres) and the right angular gyrus. They were underlined in the following publications: [Chetelat et al., 2003, Chételat et al., 2005, Drzezga et al., 2003, Morbelli et al., 2010]. To ensure that all other regions are irrelevant, we break any possible links between these regions and the output by randomly permuting all variables that belong to them. To keep the data distribution and correlation structures between and within *irrelevant* regions unchanged, we used the same random permutation vector for all features. Obviously, we have no guarantee that the four unpermuted regions are indeed truly relevant in our dataset (although most of them have been confirmed by other analysis of this dataset in this thesis), nor that all features within each of these four regions are relevant individually. We will thus have to keep this limitation in mind when analysing our experimental results below.

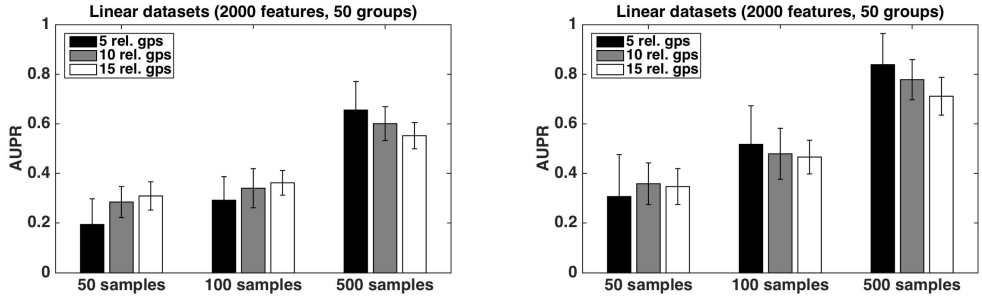
Although this dataset is not exactly real given the permutation of irrelevant groups, we will nonetheless refer to it as the *real* dataset in what follows.

Performance measures

The different feature ranking methods will be assessed by their ability to find the relevant features and to distinguish them from the irrelevant ones. Assuming that a specific threshold has been set on the estimated importance scores to decide between relevant and irrelevant features, we will use the following standard metrics to evaluate the subset of features selected according to this threshold:

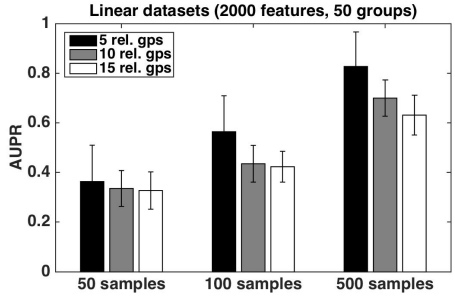
- the precision: $\frac{TP}{S}$,
- and the recall: $\frac{TP}{P}$,

where S is the number of selected features (i.e., which receive an importance score higher than the threshold), TP is the number of selected features that are truly relevant, and P is the total number of truly relevant features. To evaluate a ranking independently of the choice of an importance threshold, we will use the area under the precision-recall (AUPR) curve, with the later curve obtained by plotting precision as a function of recall for values of the importance threshold ranging from its maximum value (no feature are considered to be relevant) to 0 (all features are considered to be relevant). AUPR belongs to $[0, 1]$, with 1 corresponding to a perfect ranking with all relevant features at its top.

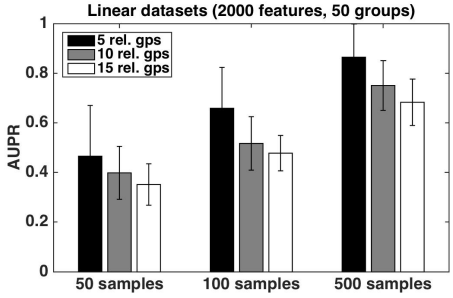


(a) $K = 1$ and $T = 200$.

(b) $K = 1$ and $T = 1000$.



(c) $K_d = \sqrt{m}$ and $T = 200$.



(d) $K_d = \sqrt{m}$ and $T = 1000$.

Figure 5.2 – Artificial datasets. AUPRs of Random Forests feature ranking method. The AUPR values are averaged over 20 datasets in each case.

5.3 Baseline

In order to properly estimate the improvement provided by the new approaches proposed in this chapter, we first analyse baseline results in this section, i.e. results obtained by standard importance scores, first on the artificial problems and then on the (pseudo-)real dataset. In order to see the impact of T and K on the improvement gained by the different approaches proposed in this chapter, we build forests of increasing sizes ($T \in \{200; 1000\}$ on the artificial dataset and $T \in \{1,000; 10,000; 100,000\}$ trees on the real datasets) and we study two settings of K, the extreme case $K = 1$ and the default case of K denoted here as $K_d = \sqrt{\#features}$. We are interested in particular to see whether the proposed approaches are more effective in the highly unstable, but more computationally efficient, settings, i.e., T and/or K small.

Artificial datasets

We generated twenty artificial datasets with 2000 features and 50 groups each, according to the model described above. We consider different numbers of relevant groups $R \in \{5, 10, 15\}$ and different number of instances $n \in \{50, 100, 500\}$. This will allow us to study the impact of both the proportion of relevant features and the ratio $\frac{\#features}{\#samples}$. Figure 5.2 shows AUPR values averaged over the twenty datasets for the different configurations of K, T, n, and R.

As expected, whatever the setting, the performance increases significantly with the number of samples. The number of relevant groups has also an impact. More precisely, the more relevant features there are, the more difficult it is to find them according to the AUPR, except in the most difficult settings (K, T, and n small). Note however that this

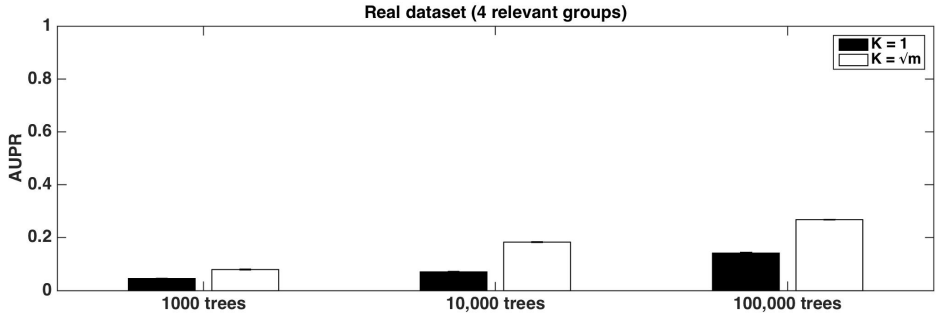


Figure 5.3 – Real dataset. AUPRs of Random Forests feature ranking method. The AUPR values are averaged over 10 runs.

comparison is purely illustrative. Indeed, AUPR values can not be strictly compared when the proportion of relevant and irrelevant features is changed, as this affects the sets and ranges of possible precision and recall values.

Comparing Figure 5.2(b) (resp. 5.2(d)) with Figure 5.2(a) (resp. 5.2(c)) shows that increasing T from 200 to 1000 leads to better performance. There is an improvement in all settings, although it is more important when either $K = 1$ or the learning sample is small, as expected. This shows that 200 trees are not enough for this problem and we will be mostly interested below in improving results in this setting.

Real dataset

We carried out the same analysis on the real dataset. Figure 5.3 shows AUPR values for the two settings of K (1 and K_d) and for three ensemble sizes ($T = 1000$, $T = 10,000$ and $T = 100,000$). AUPR values are averaged over ten runs of random forest construction.

AUPR values are globally very low on this dataset, in particular in comparison with the artificial one. The ratio between the number of features and the number of samples is much more unfavourable than on the artificial dataset. In addition, one has to keep in mind that the labelling of the features as relevant/irrelevant is also not perfect. As for the artificial dataset, K_d provides better AUPR values than $K = 1$ and increasing T also significantly improves performance. Below, we will mostly be interested in improving performance for $T = 1000$ that corresponds to the less favourable setting.

5.4 Preprocessing methods

In this section, we aim at finding the relevant groups, and so the relevant variables composing them, after having performed some preprocessing stage of the input data, leaving all other steps of the variable importance score computation unchanged. We propose to explore two preprocessing methods targeting the improvement of the traditional rankings obtained with Random Forests: *Atlas based averaging* and *Neighbourhood based averaging*. The first approach assumes some prior knowledge of the group of features, while the second one only assumes that contiguous features should contain similar information about the output. We will compare both of them with the baseline results.

5.4.1 Atlas based averaging

In this preprocessing approach, we assume that we have the knowledge of an atlas, i.e. a partitioning of the input features into non-overlapping groups of (contiguous) features. The proposed preprocessing then simply consists in using a new set of features, one for each group, computed as the average of the values of the features in the group. One then trains a Random Forests model on the learning sample with this new feature set and derive from this forest an importance score for each group. The final stage is then to attribute to each original input feature the importance score of the group it belongs to. This procedure is summarized in Algorithm 2. Note that this approach is expected to perform well in the case of the artificial dataset as averaging is a way to denoise the group features so as to get an estimate of the hidden variable z corresponding to each group.

Algorithm 2 Atlas based averaging

Require: Learning sample LS, algorithm $\mathcal{RF}$ to obtain importance scores from a forest, group division $(G_1, \ldots, G_g)$ of the features

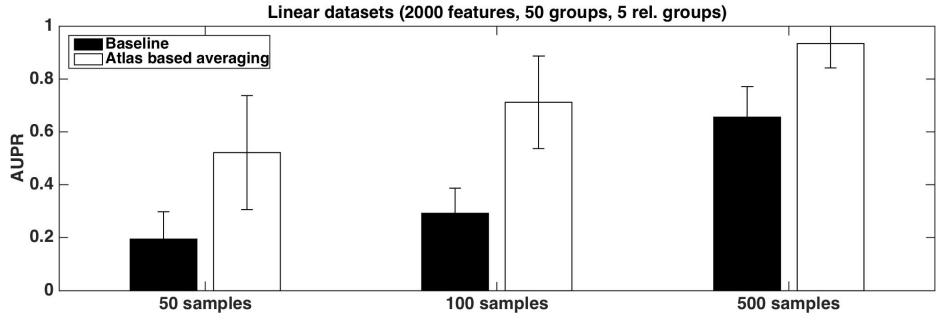
// Compute the g new features as follows:
for $i = 1 : g$ **do**
    $x_i^{new} = \frac{1}{\#G_i} \sum_{x_j \in G_i} x_j$
end for
Let LS^{new} be LS where original features are replaced by new features $(x_1^{new}, \ldots, x_g^{new})$.
Compute variable importance scores $(s_1, \ldots, s_g) = \mathcal{RF}(LS^{new})$.
Attribute s_i to all features $x_j \in G_i$ for $i = 1$ to g.

The idea of region-based feature averaging has been explored in a couple of publications. In [Pagani et al., 2015], they exploited this idea for the discrimination of MCI patients about to convert and healthy controls from FDG-PET scans. Input features of their learning machine were specifically the average intensity of volumes of interest defined by an anatomical atlas. Their experiments showed good accuracies in classification. Gray et al. [2012] built also SVM-based classifiers in their paper with a similar procedure, averaging regional intensities of FDG-PET images for the prognosis of MCI patients. Although this method appears to provide good results, they noted that such averaging approach reduces the effective image resolution and prevents the fine analysis of brain patterns linked to a patient condition.

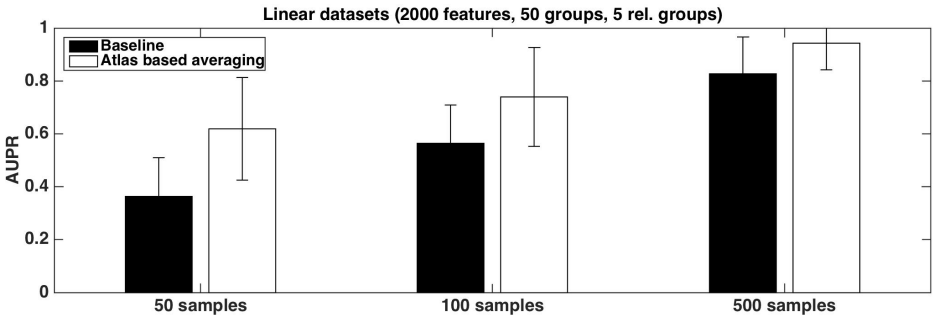
Artificial datasets

In the following experiment, we focus exclusively on the case of 5 relevant groups. We have made additional experiments (not shown) confirming that methods behave in a similar way for 10 and 15 relevant groups.

Figure 5.4 compares AUPRs of Random Forests rankings with and without preprocessing for $T = 200$. The group division used for atlas based averaging is the one used to create the artificial dataset. AUPRs for this latter method in Figures 5.4(a) and 5.4(b) are thus computed from feature rankings in which all features belonging to the same group have the same relevance score. Comparison between the baseline and the preprocessing alternative are performed through Figure 5.4(a) for $K = 1$ and through Figure 5.4(b) for K_d, where K_d is equal to $\sqrt{m}$ for the baseline and to $\sqrt{g}$, with g the number of groups, for atlas based averaging. Figure 5.5 illustrates the effect of atlas based averaging on



(a) $K = 1$.



(b) K_d.

Figure 5.4 – $T = 200$. Atlas based averaging on the artificial datasets. AUPRs of Random Forests feature ranking method. The group division for preprocessing is the one used to create the dataset. The AUPR values are averaged over 20 datasets in each case.

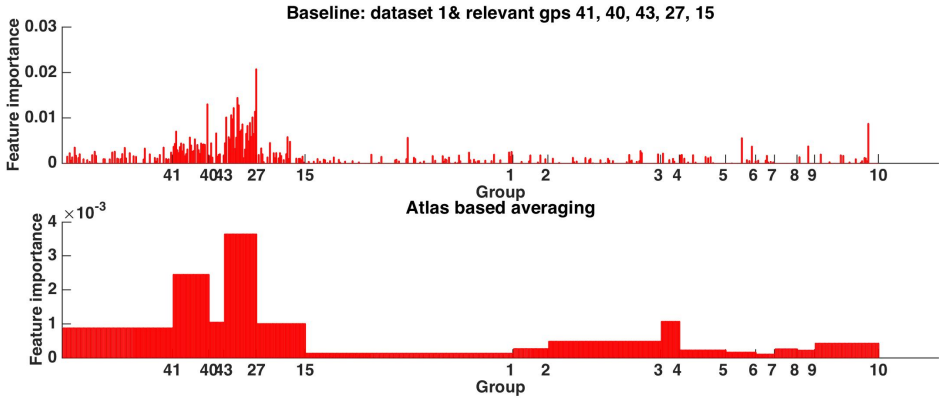


Figure 5.5 – $T = 200$ and K_d. Atlas based averaging on the first artificial dataset for 50 samples. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

importance scores on the first artificial dataset.

As shown in Figure 5.4, the preprocessing procedure improves AUPR values both for $K = 1$ and K_d. Interestingly, analysing the figures, atlas based averaging seems to

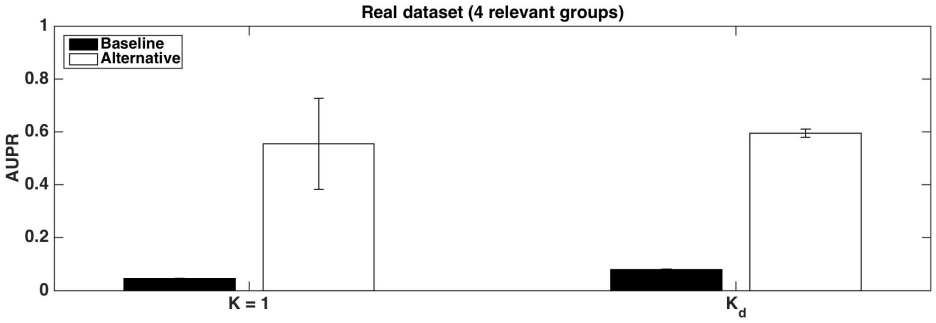


Figure 5.6 – $T = 1000$. Atlas based averaging on the real dataset. AUPRs of Random Forests feature ranking method. The group division for preprocessing is given by the AAL atlas. The AUPR values are averaged over 10 runs.

provide very similar AUPR values both for $K = 1$ and K_d, although the latter setting is superior in the case of the baseline. Indeed, there is not a large difference between the two settings: AUPRs are 0.52, 0.71 and 0.93 respectively for 50, 100 and 500 samples with $K = 1$, and 0.62, 0.74 and 0.94 respectively for 50, 100 and 500 samples with K_d. There seems to be a significant difference between the two AUPR values only for the smallest sample size.

Real dataset

Figure 5.6 compares the baseline and atlas based averaging for $T = 1000$ on the real dataset. We used the same (AAL) atlas for feature value averaging as for the selection of relevant groups explained in Subsection 5.2. As we can see, AUPR values are increased both for $K = 1$ and K_d in comparison with the baseline. As for the artificial datasets, averaging over the groups seems to make the method more robust to the value of K. We however observe a much higher variance with $K = 1$ than with K_d, which is not surprising as $K = 1$ involves more randomization.

5.4.2 Neighbourhood based averaging

The main drawback of atlas based averaging is that it requires knowledge of the group structure. The alternative preprocessing technique proposed in this section tries to circumvent this issue. The approach only assumes that the group structure is consistent with some given spatial organization of the features.

The general idea of the approach, called *neighbourhood based averaging*, is very similar to the previous atlas based averaging approach except that instead of using a pre-defined partition of the features in non-overlapping groups, it relies on a function $\mathcal{G}$ that will generate a potentially large set of z groups of features. These groups are expected now to be overlapping and to be consistent with the neighbourhood relationship defined on the features, i.e., to be composed only of features that are spatially contiguous. From these groups, the algorithm computes a new learning sample with a new feature for each group computed by averaging the values of the original features in the group. Importance scores are then derived for each group from a Random Forests model trained on the new input matrix and these importance scores are mapped back to each original feature by computing the average importances over the groups to which that feature belongs. The procedure is described more formally in Algorithm 3.

Unlike atlas based averaging, the approach does not require any a priori information about the exact group structure. The idea is that if relevant groups are included among (or close to some of) the groups generated through the function $\mathcal{G}$, the random forests algorithm will be able to identify them among all the other candidate groups. It requires however to define the group generator $\mathcal{G}$, which typically will depends on some hyper-parameters. We propose and experiment below with two group generators specifically designed for the artificial and real datasets respectively.

Algorithm 3 Generic neighbourhood based averaging algorithm

Require: Learning sample LS, algorithm $\mathcal{RF}$ to obtain importance scores from a forest, group division generator $\mathcal{G}$ of size z.

1: Generate groups $(G_1, \ldots, G_z) = \mathcal{G}(LS)$.
2: **for** $j = 1 : z$ **do**
3: $\quad x_j^{new} = \frac{1}{\#G_j} \sum_{x_k \in G_j} x_k$
4: **end for**
5: Let LS^{new} be LS where original features are replaced by new features $(x_1^{new}, \ldots, x_z^{new})$.
6: Compute variable importance scores $(s_1^g, \ldots, s_z^g) = \mathcal{RF}(LS^{new})$.
7: Compute importance s_i of original feature x_i (for $i = 1, \ldots, m$) as follows:

$$s_i = \sum_{k \in \{1,\ldots,z\} | x_i \in G_k} s_k^g.$$

8: Divide s_i by the number of different groups to which feature x_i belongs, i.e.,

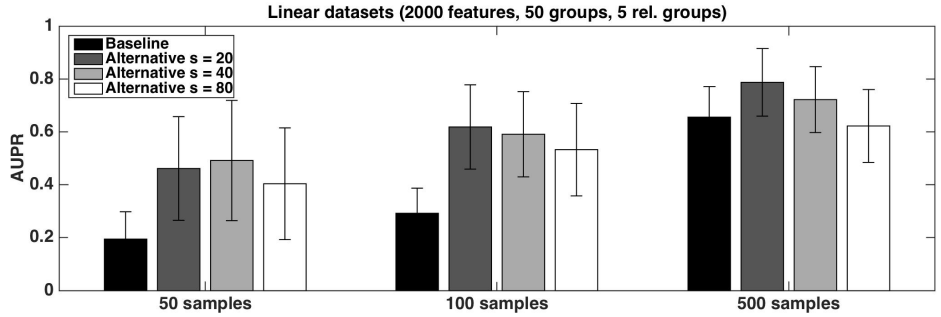
$$|\{k \in \{1, \ldots, z\} | x_i \in G_k\}|.$$

Artificial datasets

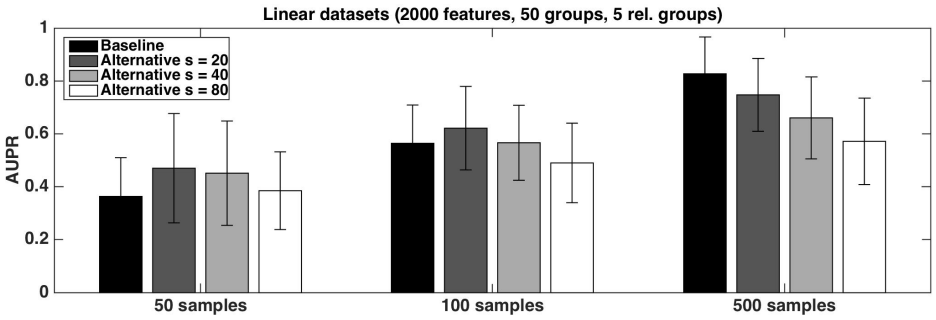
In the artificial datasets, there is a linear organization of the features, since the groups correspond to blocks (of variable sizes) of features along their original ordering. The group generator function that we will consider generates groups in the form of blocks of contiguous features of fixed size along their ordering. We set block size to $s + 1$ with s even so that each block can be considered as centred on one of the original features and contains this feature and its s closest neighbours in the ordering. If there are m features originally, then the number of generated groups will be $z = m - s$.

We analyse different neighbourhood sizes s in Figure 5.7 for $T = 200$, with $K = 1$ in Figure 5.7(a) and K_d in Figure 5.7(b) (with $K_d = \sqrt{m - s}$ in the case of neighbourhood based averaging). The tested sizes are 20, 40 and 80 where 40 corresponds to the average size of a true group by construction of the artificial datasets (50 groups for a total of 2000 features). Figure 5.8 illustrates the effect of the pre-processing on importances scores on the first artificial dataset (with K_d, 50 samples, and $s = 20$).

In Figure 5.7(a), we observe that, for $K = 1$, the procedure provides better AUPR values than the baseline in almost all settings. Notably for $s = 20$, the preprocessing approach always beats the baseline, while it does not perform better than the baseline with $s = 80$ and 500 samples. The value of s has an observable impact on the efficiency of the method, with the best performance depending on the number of samples. For K_d in Figure 5.7(b), neighbourhood based averaging improves the results for all values of



(a) $K = 1$.



(b) K_d.

Figure 5.7 – $T = 200$ and $z = m - s$. Linear neighbourhood based averaging on the artificial datasets. AUPRs of Random Forests feature ranking. The AUPR values are averaged over 20 datasets in each case.

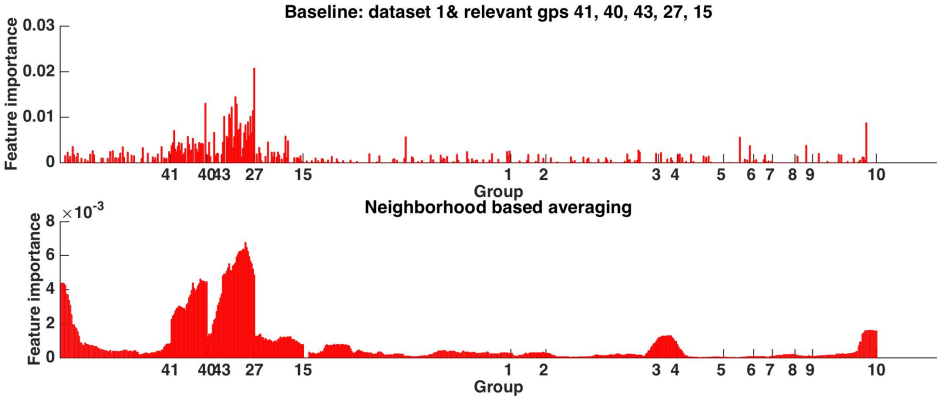


Figure 5.8 – $T = 200$, K_d. Linear neighbourhood based averaging on the first artificial dataset for 50 sample and $s = 20$. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

s only with 50 samples. It still improves for 100 samples when $s = 20$ and $s = 40$ but it deteriorates performance in all other settings.

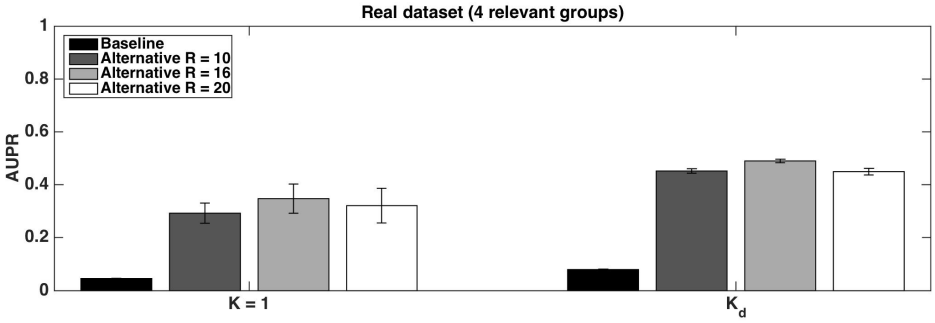


Figure 5.9 – $T = 1000$. Spherical neighbourhood based averaging on the real dataset. AUPRs of Random Forests feature ranking. The AUPR values are averaged over 10 runs.

Real dataset

In the real dataset, spatial organization of the features is three dimensional. Therefore, groups should be generated in a different way. We get for this inspiration from the *searchlight* approach used in the field of fMRI [Kriegeskorte et al., 2006]. This multivariate pattern analysis method consists in the evaluation of the information contained in spherical volumes centred on every voxel of the brain [Kriegeskorte et al., 2006]. In particular, the method associates to each voxel the capability of its searchlight region to distinguish the studied conditions. This results in a statistical map in which each value brings interpretation relative to a sphere of voxels instead of an individual voxel itself.

Inspired by this method, our group generator $\mathcal{G}$ creates groups of the atlas by associating to each feature a sphere of radius R centred on this feature. The amount of overlap between the groups directly depends on the value of R. In this configuration, the new input matrix has as many features as originally. We propose to evaluate three distinct settings: $R = 10\ mm$, $R = 16\ mm$, and $R = 20\ mm$. They respectively correspond to spheres composed of around 470, 1830, and 3480 features. These values have been chosen to explored neighbourhood of sizes lower, close, and greater than the average group size in the AAL atlas (ie., 1431 features. See Appendix C).

Results are shown in Figure 5.9. We observe that the neighbourhood based averaging improves AUPR values compared to the baseline for any value of R. The case $K = 1$ shows higher variance than K_d. The value of R influences the performance of the approach and the best AUPR value is obtained for $R = 16$ both for $K = 1$ and K_d. It is worth to note that K_d both for the baseline and for the alternative procedures corresponds to $K = \sqrt{m}$, as each feature value is replaced by the average feature values in a sphere centred on it.

5.4.3 Discussion

Although results are very good for the atlas based averaging procedure, it is quite rare in general to perfectly know the groups to which the relevant and irrelevant features belong. Indeed, in neuroimaging, we often work with atlas defining brain division into several groups according to anatomical or functional consideration. That kind of features separation links a particular voxel to a brain area and thus helps for the interpretation of the role of the variables highlighted by methods. Although that type of atlas is available to interpret results, relevant groups of features will not necessarily match

with an entire brain area. Relevant groups of voxels can correspond to small fractions of brain areas or can be distributed over two adjacent regions.

We therefore proposed another approach called Neighbourhood based averaging. This approach is independent of an atlas a priori defined. Both for artificial datasets and real dataset, it provided good improvements compared to the baseline for both K values and for different atlas sizes. Although the choice of s or R can be difficult in practice, this approach seems promising. The approach is furthermore generic.

5.5 Embedded methods

In this subsection, we explore two modifications of Random Forests algorithm aiming at improving the feature importance scores it provides for a given number of trees T. We call these methods embedded because they involve modifications of the Random Forests algorithm itself.

5.5.1 Sum of potential node impurity decreases

For high dimensional data, the risk still remains to not attribute an importance score to a feature even if it has been seen during the learning process. Indeed, seen does not mean chosen for $K > 1$ (cf. Chapter 4). For each node of a tree, the impurity reduction will be computed for K different features, but only the best feature, in terms of impurity reduction, will be selected to split the node and hence yield a non-zero contribution to its importance for this node. In the standard method, all these potential impurity reductions of the unselected features are thus disregarded. In this section, we propose to exploit these potential impurity reductions in order to obtain importance scores for all the features, even those which have not been selected.

The idea is to save at each node the decrease of impurity computed for each of the K features and to derive the importance of each variable by summing all the impurity decreases computed for each node where this variable was evaluated and not only those where it has been selected as the best splitting variable. The modification of the function to learn a randomized tree with respect to the standard Random Forests algorithm is shown in Algorithm 4 (changes with respect to Algorithm 1, page 18, are highlighted in red). This algorithm is also studied in Antonio Sutera's thesis as this is a collaborative work.

This algorithm has no influence when $K = 1$ as in this case, a single feature is considered at each node and it is used to split the node whatever the impurity reduction. When $K > 1$, the hope is that it will increase the number of evaluations of each feature taken into account in the computation of its importance and therefore have an effect similar to an increase of the forest size that will improve stability. The importance scores so computed will however converge to different values as the original scores when T increases, which might affect the performance in one way or another.

Artificial datasets

Figure 5.10 reports the AUPR values obtained with the standard Random Forests algorithm in comparison with the ones obtained with the sum of potential node impurity decreases for K_d only (as with $K = 1$, the modification has no effect). We observe only a very slight improvement of AUPRs overall. While this is disappointing, this result is however somehow consistent with the fact that on these datasets, increasing T beyond 200 only slightly increases AUPRs (see Figure 5.2). Figure 5.11 shows the impact of the

Algorithm 4 Sum of potential node impurity decreases

function LEARN_A_RANDOMIZED_TREE_ACC(LS)
  if all objects from LS have the same class **then**
    Create a leaf with that class.
  else
    Randomly pick K features.
    for each feature x among K **do**
      Evaluate the expected reduction of impurity $\Delta I(\mathcal{N})$ provided by the best split on x
      $\mathcal{I}(x) \leftarrow \mathcal{I}(x) + n_{\mathcal{N}} \Delta I(\mathcal{N})$.
    end for
    Select the feature x^* giving rise to the maximum $\Delta I(\mathcal{N})$.
    Create a test node for the selected split and divide LS into sub-samples LS_1 and LS_2 according to this split.
    LEARN_A_RANDOMIZED_TREE_ACC(LS_1)
    LEARN_A_RANDOMIZED_TREE_ACC(LS_2)
  end if
end function

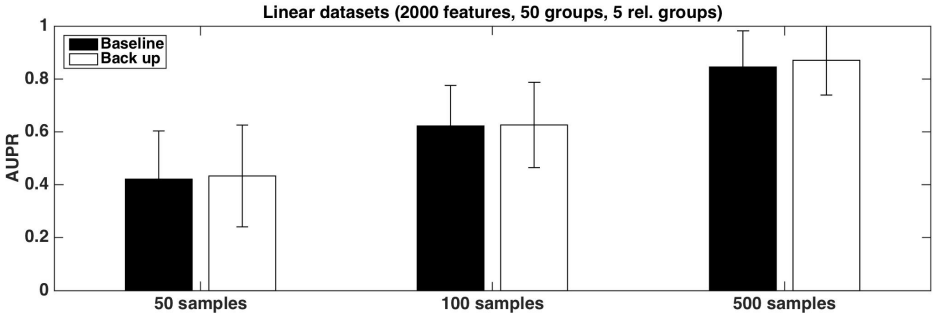


Figure 5.10 – $T = 200$ and K_d. Sum of potential node impurity decreases on the artificial datasets. AUPRs of Random Forests as feature ranking methods. The AUPR values are averaged over 20 datasets in each case.

method on the importance score distribution in the case of the first artificial dataset. As expected, the approach leads to an overall increase of importance scores, with much less features with a zero importance for the same number of trees.

Real dataset

On the real dataset, Figure 5.12 shows a larger increase of the average AUPR value over the ten runs. For a database with so many features, it seems thus advantageous to exploit all the potential decreases of impurity computed along the learning process.

5.5.2 Group Random Forests

Another potential issue with standard Random Forests appears for structured data when it comes to highlight groups instead of features and when these groups are of potentially very different sizes. In standard Random Forests, each feature has the same chance to be among the K features randomly drawn at each tree node. Consequently,

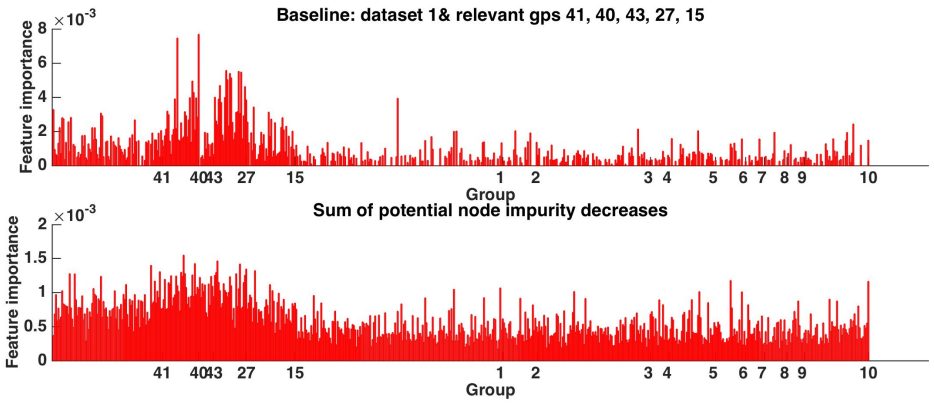


Figure 5.11 – $T = 200$ and K_d. Sum of potential node impurity decreases on the first artificial dataset for 50 samples. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

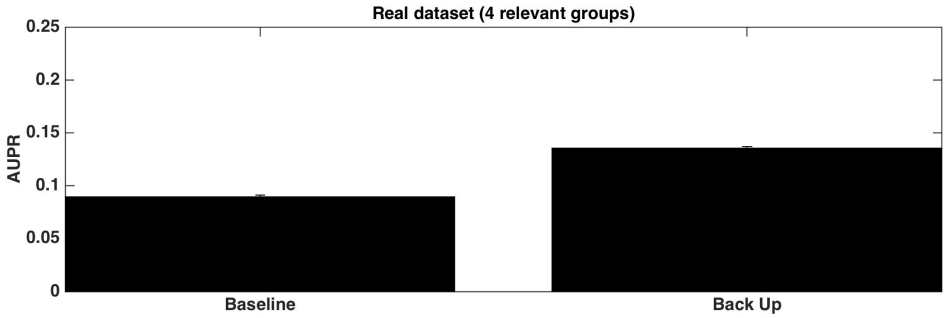
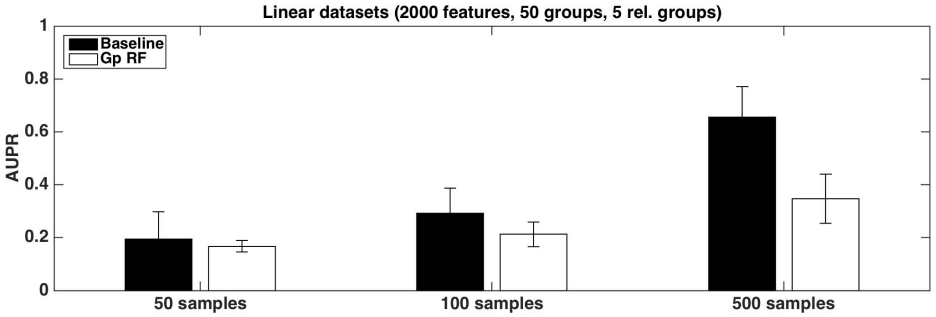
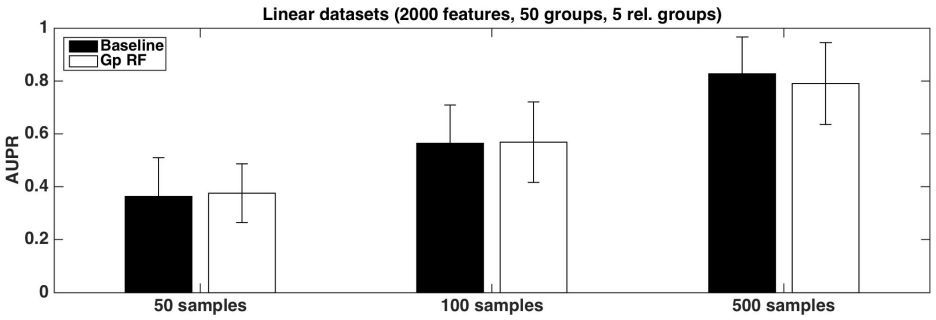


Figure 5.12 – $T = 1000$ and K_d. Sum of potential node impurity decreases on the real dataset. AUPRs of Random Forests as feature ranking methods. The AUPR values are averaged over 10 runs.

there is a higher probability to select a feature from a large group than from a small one. A potentially negative consequence is that larger irrelevant groups will have more chance to have one of their features to be selected by chance than maybe smaller relevant groups. We propose to address this issue by changing the way variables are randomly picked in order to obtain a procedure which is more fair with respect to group sizes. The procedure works as follows: at each node, K groups are randomly drawn, with replacement, and one feature is randomly picked in each of these groups. We call this modified RF algorithm *Group Random Forests*. The groups are drawn with replacement to be able to set K independently of the number of groups. If K is larger than the number of groups, then several features will be simply drawn from some of the groups. With this modified procedure, groups will be represented equally in the selected features. For a given feature x_i, the probability that it gets selected will be modified from $\frac{1}{m}$ with standard Random Forests to $\frac{1}{g} \times \frac{1}{\#G_i}$ in the new procedure, with g the number of groups and $\#G_i$ the size of the group that contains x_i. Features inside large groups will thus have a lower probability to be selected. This procedure implicitly assumes that features in one group are interchangeable, as it prefers to explore more groups than to explore more features within large groups.



(a) $K = 1$.



(b) K_d.

Figure 5.13 – $T = 200$. Group Random Forests on the artificial datasets. AUPRs of Random Forests and Group Random Forests as feature ranking methods. The AUPR values are averaged over 20 datasets in each case.

Artificial datasets

Random Forests rankings and Group Random Forests rankings are compared in Figure 5.13 for $K = 1$ and K_d. For $K = 1$, Group Random Forests clearly decrease the variance of the AUPR over the different datasets. Unfortunately, we do not observe an increase of AUPR values by the group approach. On the contrary, there is even a considerable loss of AUPR, especially for 500 samples. There is also no improvement with K_d, but Group Random Forest is only marginally worse for this setting. Figure 5.15 illustrates the impact of the method on the importance scores in different groups for the first artificial dataset. From this figure, one can see that importances of features in small groups are reinforced (e.g., groups 2 and 4 among the irrelevant ones and groups 40 and 27 among the relevant ones), while importance scores are more scarce for larger groups (e.g., group 1). This however does not translate into an improvement of AUPR for these problems.

To analyse more deeply the influence of this method on the importance scores, we computed the median, mean and sum values of the importance scores inside each relevant group both with Group RF and standard RF. Figure 5.15 plots the difference between values obtained with a standard Random Forests and those obtained for Group Random Forests as a function of group size. As expected, these differences are negative for groups of smaller sizes, since features from small groups have a higher probability to be seen in the case of Group RF. For groups of size larger than 40 (which is the average size of a group), the behaviour is the opposite.

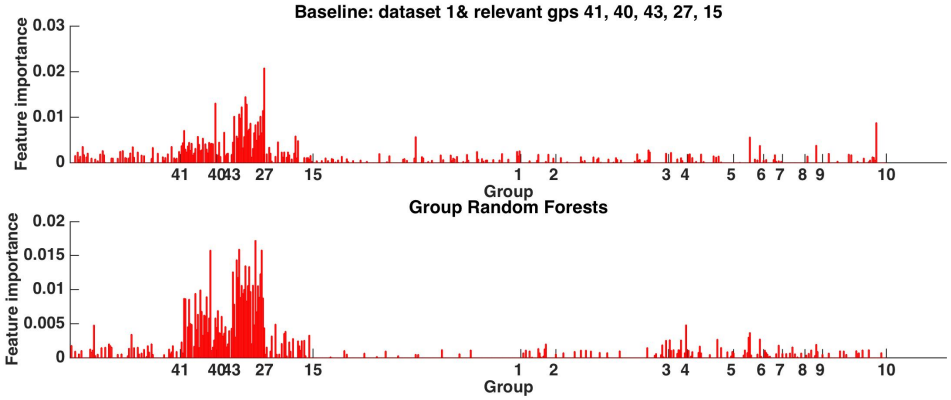


Figure 5.14 – $T = 200$ and K_d. Group Random Forests on the first artificial dataset for 50 samples. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

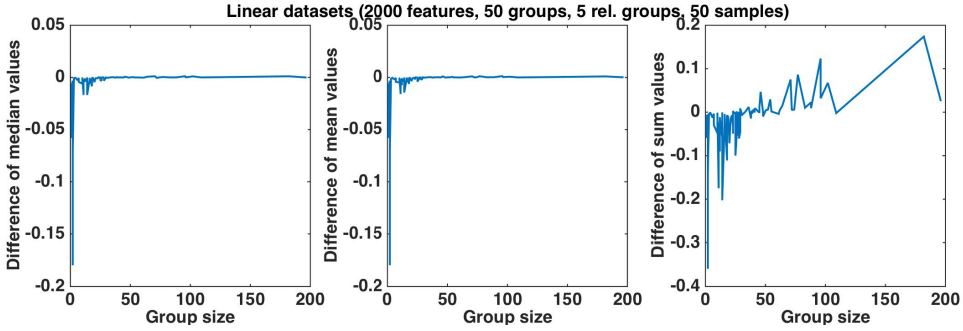


Figure 5.15 – $T = 200$. Group Random Forests on the artificial datasets. Comparison of median, mean and sum values of importance scores inside relevant groups depending on the size of the relevant group (Random Forests value-Group Random Forests value).

Real dataset

We observe in Figure 5.16 similar AUPR values for $K = 1$ and a slight decrease for K_d. Actually, on this dataset, only one configuration of groups is considered and the relevant groups are relatively large compared to the irrelevant ones: their sizes exceed the median of the group sizes. Consequently, favouring smaller groups is not expected to improve on this dataset.

5.5.3 Discussion

We have proposed two different adaptations of the Random Forests algorithm. The first one directly impacts the importance scores by cumulating all decreases of impurity computed, while the second one modifies the probability of randomly picking a variable so as to enforce a more fair treatment of groups during the learning process. None of these approaches have provided really promising results in the case of the artificial datasets. Nevertheless, the *Sum of potential node impurity decrease* approach provides an improvement of the feature rankings in the case of the real dataset, although not at level of what was gained with the pre-processing techniques.

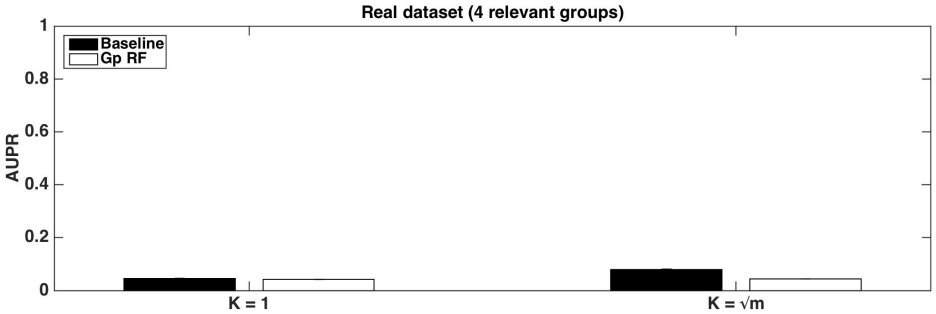


Figure 5.16 – $T = 1000$. Group Random Forests on the real dataset. AUPRs of Random Forests and Group Random Forests as feature ranking methods. The AUPR values are averaged over 10 runs.

5.6 Postprocessing methods

In this section, we focus on postprocessing methods. The main idea of these methods is to collect feature importance scores by applying the standard RF method on the original data matrix and then to postprocess them a posteriori to improve their interpretability and usability. As for the preprocessing techniques, we consider two approaches, one that only takes into account the spatial ordering of the features (called *Neighbourhood based smoothing*) and one that takes into account (and thus requires) a pre-existing group structure (called *Group based aggregation*).

5.6.1 Neighbourhood based smoothing

The smoothing approach aims at attributing an importance score to all the features starting from a sparse importance score vector. With a too small forest learnt on high dimensional data, some features are not observed at all, while others are seen in only very few configurations. Their importance values can therefore not be estimated reliably. The idea of the approach proposed here is to spatially smooth the importance scores. This idea is motivated by the hypothesis that features in the neighbourhood of an important feature should be relatively important too. Under this hypothesis, if one feature receives some importance because it is selected more often than its neighbours by chance or because it masks them, then it makes sense to share some of its importance with these neighbours. Algorithm 5 describes this general procedure.

Algorithm 5 Neighbourhood based smoothing

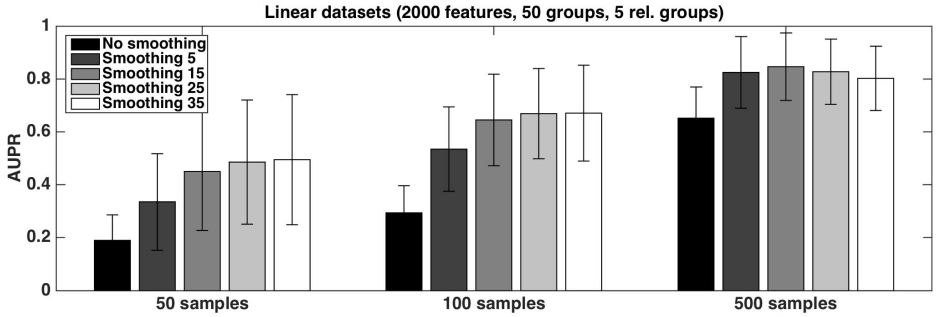
Require: LS, $\mathcal{RF}$, and a smoothing operator $smooth_z$ with parameter z.

1: Compute importance scores of all features:

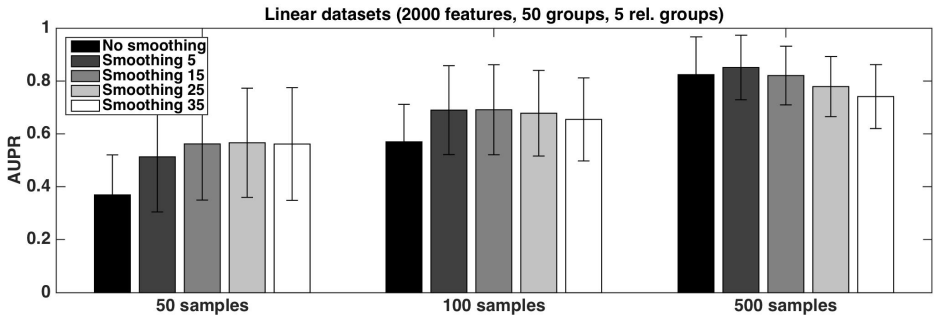
$$(s_1, \ldots, s_m) = \mathcal{RF}(LS).$$

2: Apply smoothing operator on these importance scores:

$$(s_1, \ldots, s_m) \leftarrow smooth_z \left\langle (s_1, \ldots, s_m) \right\rangle .$$



(a) $K = 1$.



(b) K_d.

Figure 5.17 – $T = 200$. Neighbourhood based smoothing on the artificial datasets. AUPRs of Random Forests group ranking method. The AUPR values are averaged over 20 datasets in each case.

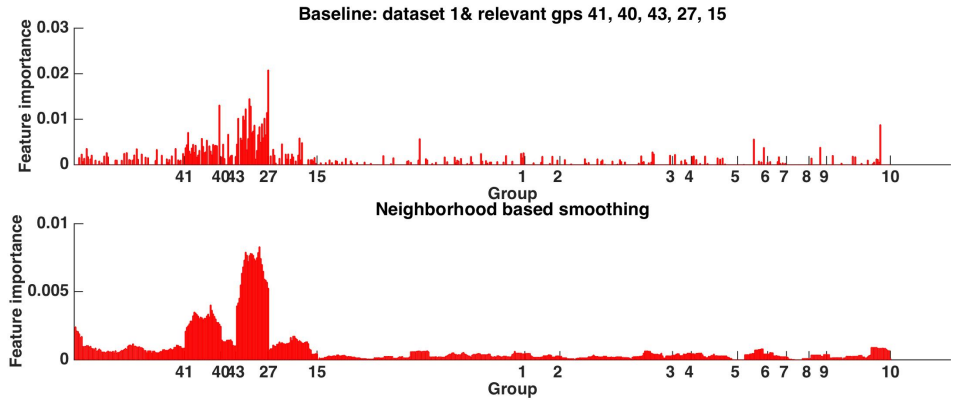


Figure 5.18 – $T = 200$ and K_d. Neighbourhood based smoothing on the first artificial dataset for 50 samples. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

Artificial datasets

For the artificial datasets, we evaluate a simple moving average smoothing filter. The unique parameter of that filter is the size of the subwindows (in terms of number of features) over which the average is computed. Results with different subwindow sizes

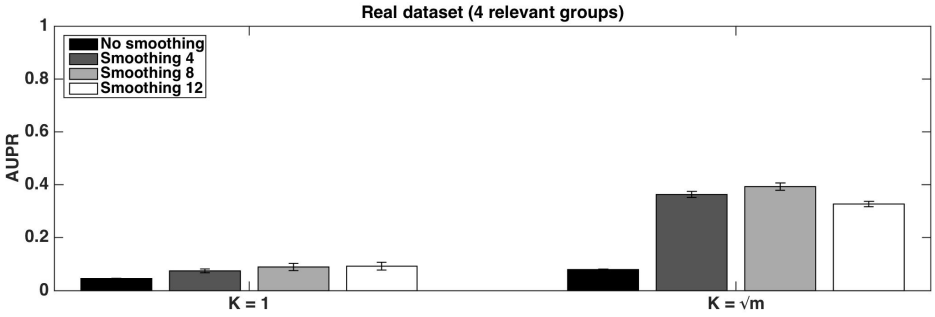


Figure 5.19 – $T = 1000$. Neighbourhood based smoothing on the real dataset. AUPRs of Random Forests feature ranking method. The AUPR values are averaged over 10 runs.

are shown in Figure 5.17, while Figure 5.18 illustrates the impact of smoothing on importance scores in a few groups on the first artificial dataset.

For $K = 1$, in Figure 5.17(a), we observe an important improvement of AUPR values with the smoothing operation. The larger subwindow size seems to be well adapted to the smaller learning sample sizes, 50 and 100. This makes sense as smaller learning samples are expected to produce more sparse importance vectors. With a sample size of 500, AUPR slowly decreases when the subwindow size is greater than 15. Although the increase is less impressive for K_d (Figure 5.17(b)), it is still noticeable for 50 and 100 samples in particular. For this setting, as for $K = 1$, more smoothing has to be applied for smaller learning sample sizes.

Real dataset

On the real dataset, we evaluate a spatial gaussian smoothing similar to the one performed in SPM for image preprocessing. The parameter of the smoothing is the full width half maximum (FWHM), which has been set to 4, 8, and $12mm$. We observe in Figure 5.19 that the intermediate parameter value, $FWHM = 8mm$, is the best for K_d. For $K = 1$, differences are not sufficiently large to conclude in favour of one parameter value compared to another. Nevertheless, for both settings of K, we observe that such postprocessing improves significantly in terms of AUPR.

5.6.2 Group based aggregation

The second approach we proposed in this section consists in inferring group importance scores by aggregating the individual importance scores of the features in the group. This idea has been proposed in [Schrouff et al., 2013a] for SVM weights, where the proposed aggregation operator was averaging. This technique is based on the principle that, in the context of neuroimaging data, we are looking for groups of voxels instead of isolated features. Because of this, we will use feature importance scores for the information they can provide about groups of features. Once we have an importance score for each group, we attribute to each feature an importance score equal to the importance of the group in which it belongs. We provide in Algorithm 6 a pseudo-code for this approach.

We investigate three different aggregation functions of individual importance scores: the mean, the sum and the max. Louppe et al. [2013] have shown that the sum of the MDI importances of all features represents the total amount of class impurity reduction brought by the forest. Taking the sum of the importances is thus the most natural

Algorithm 6 Group based aggregation of importance scores

Require: LS, $\mathcal{RF}$, group division $(G_1, \ldots, G_g)$ of the features in g groups, aggregation function $\mathcal{A}$.

1: Compute variable importance scores $(s_1, \ldots, s_m) = \mathcal{RF}(LS)$.
2: **for** $i = 1$ to g **do**
3: $\quad Imp(G_i) = \mathcal{A}(\{s_j | x_j \in G_i\})$.
4: **end for**
5: Attribute $Imp(G_i)$ to all features $x_j \in G_i$ for $i = 1, \ldots, g$.

choice: the importance of a group is the total class impurity reduction brought by the features from the group. The sum has however the drawback that it is potentially biased towards groups of larger size. Indeed, large groups have more chance to have their features selected when building the forest. The average avoids any bias due to differences in group cardinality but has the drawback that a group can not be important if only a small proportion of its features are important. Finally, taking the maximum of the importances in the group assumes that the feature of highest importance alone is representative of the group importance. As it is unclear a priori which aggregation function would work best in practice, we will compare all of them on both the artificial and real datasets.

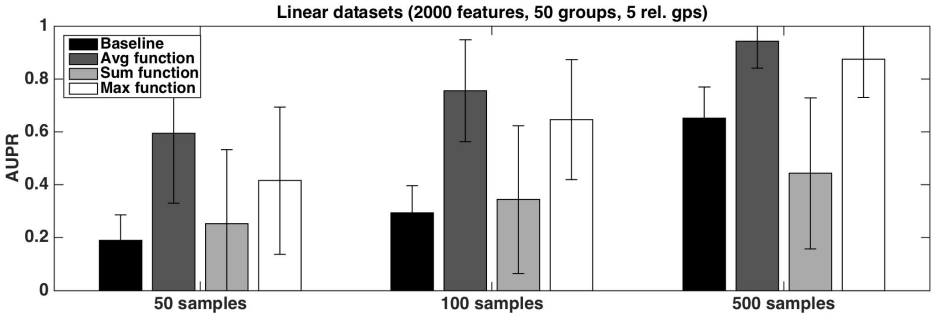
Artificial datasets

In order to see if such approach is of interest, we analyse the AUPR values for each aggregation function. Figure 5.20 shows that the best aggregation function is the average for both K values. This aggregation function improves considerably the baseline AUPR, whatever the value of K and the number of samples. The improvement is more important for smaller sample sizes. The behaviour of other aggregation functions seems to depend on K. For $K = 1$, the sum is clearly the worse performer while it is better or similar to the max aggregation for $K_d = \sqrt{m}$. For 50 samples with K_d, the sum is better than the max while for higher sample sizes, the two methods perform similarly. However, for small sample sizes, all aggregation functions provide an improvement for both K values.

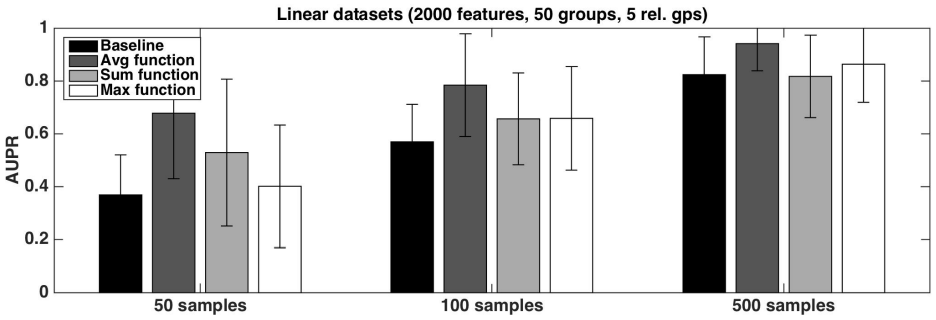
As for the other methods in this chapter, Figure 5.21 illustrates the impact of group based aggregation on a few groups on the first artificial dataset. The comparison between the three aggregation operators is interesting, and it explains why the average operator works best on this problem. With the average operator, as expected, the first five groups, which are relevant, receive a higher score than the next ten groups, which are irrelevant. With the sum operator, the irrelevant group 1, while it contains mostly low score features, nevertheless receives a higher score than the relevant group 43, while it contains higher score features. This happens only because the former is larger than the latter and the sum is biased in favour of larger groups. Finally, with the max operator, several irrelevant groups (1, 6, and 10 for example) receive a higher score that group 41, because they contain each a single feature that receives a high score only due to noise (since they are irrelevant by construction). This method thus appears to be more prone to noise than the average.

Real dataset

In Figure 5.22, we observe that the sum is performing significantly worse than the other aggregation functions for $K = 1$. However, it shows a lower variance than others for both K values. For K_d, the max and the sum provide similar AUPR values with a considerable improvement of AUPR values compared to the baseline. The average



(a) $K = 1$.



(b) K_d.

Figure 5.20 – $T = 200$. Aggregation on the artificial datasets. AUPRs of Random Forests feature ranking method. The group division for postprocessing is the one used to create the dataset. The AUPR values are averaged over 20 datasets in each case.

provides the best improvement for $K_d = \sqrt{m}$, where group aggregation allows to reach an almost perfect ranking.

5.6.3 Discussion

Smoothing appears to improve the quality of the importance scores, in particular for smaller sample sizes. It has the advantage of not requiring a prior group division. However, it depends on a smoothing level hyper-parameters, whose value has an impact on performance and that might be difficult to tune in practice.

When a group division is known, group based aggregation performs very well both on the artificial and the real datasets. All aggregation functions brings some improvement with respect to the baseline, whatever K and the learning sample size. On the artificial datasets, we observed better results with the average function. This aggregation function also provides the best results on the real dataset.

5.7 Summary

In this chapter, we proposed several improvements of Random Forests based importance scores that exploit either some spatial organization of the features or a pre-defined division of these features into groups. One of the motivations for introducing these methods is to reduce the needs in terms of ensemble size to obtain reliable importance

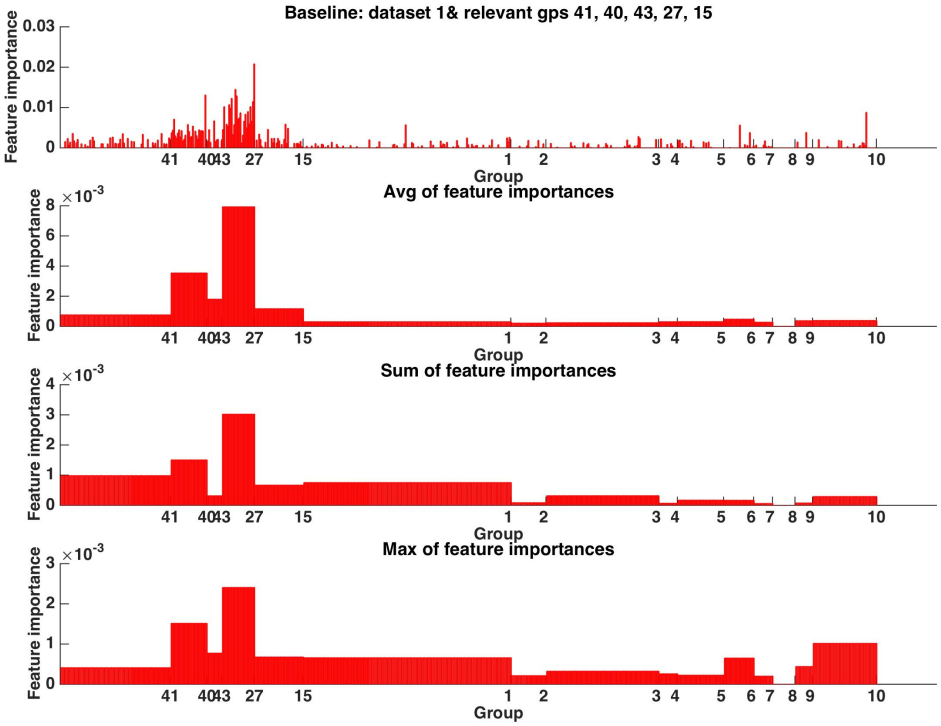


Figure 5.21 – $T = 200$ and K_d. Aggregation on the first artificial dataset for 50 samples. Distribution of importance scores. The numbers on the x-axis represent the groups in which the features belong. They are placed at the end of the group.

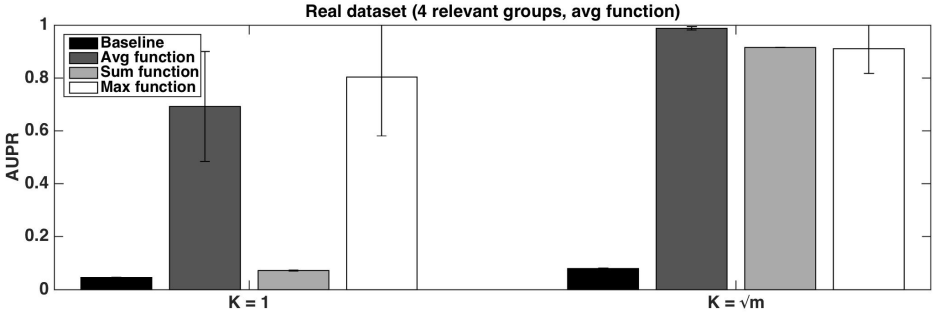


Figure 5.22 – $T = 1000$. Aggregation on the real dataset. AUPRs of Random Forests feature ranking method. The group division for postprocessing is given by the AAL atlas. The AUPR values are averaged over 10 runs.

scores in small n/large m settings. Overall, our experiments indeed confirmed that some of these alternative methods make the results less sensitive to both the number of trees T but also the value of K.

Firstly, we proposed two preprocessing approaches. The first one assumes that a group division of the features is available and simply averages features in each group,

reducing the number of features to the number of groups. This procedure works well but requires that groups are known a priori. As an alternative when groups are unknown, we proposed Neighbourhood based averaging, which appears as a sensible alternative.

Secondly, we investigated two modifications of the way Random Forests method computes importance scores. The first method accumulates the mean decrease of impurity for all features seen during the tree construction, while the second one modifies the random selection of K features at each tree node by a random selection of K groups. Unfortunately, none of these adaptations has shown a real benefit in our experiments on the artificial datasets. On the real dataset, the accumulation approach revealed a small improvement, but it is not competitive with the other approaches investigated in this chapter. Actually, this method does not exploit explicitly the structure of the data, which might explain why it is not competitive. Note that the Group Random Forests approach modifies the forest that is built and it can thus also have an impact on predictive performance. We compare this method with standard Random Forests in terms of accuracy on several datasets in Chapter 9.

Finally, we studied two post-processing approaches applied on feature importances derived by the standard Random Forests algorithm. The first approach simply smoothes the importance scores by taking into account the spatial organization of the features; it revealed to be interesting, especially on the artificial datasets, but requires to carefully tune the size of the smoothing neighbourhood. The second idea was the aggregation of importance scores over groups of features, by using different aggregation functions such as the average, the sum, and the max. This approach requires a prior knowledge of the feature groups. We obtained the best results when aggregating by the average. As a matter of fact, the improvements of AUPR values obtained in this latter case are the best among all methods proposed in this chapter. This approach will be further explored in the next two chapters of this thesis.

We summarize the performance results of all methods in Tables 5.1 and 5.2 for the artificial datasets and the pseudo-real dataset respectively. As stated above, the Group based aggregation method with the average aggregation function provide the highest improvement of AUPR values. The Atlas based averaging method closely follows these results. However, these two methods require the use of an atlas. As an alternative, the neighbourhood based averaging and the neighbourhood based smoothing can also improve the baseline AUPRs. The embedded methods did not provide significant improvements.

Table 5.1 – Comparison of method AUPR values for the artificial datasets. For neighbourhood based averaging, results displayed correspond to $s = 20$ for both K values. For neighbourhood based smoothing, we displayed the results corresponding to $z = 15$ for $K = 1$ and $z = 5$ for K_d. Group based aggregation results correspond to the average aggregation function.

		Baseline	Preprocessing		Embedded		Postprocessing	
	n		Atlas	Neighb.	$\sum \Delta I$	Gp RF	Smooth.	Gp aggr.
$K = 1$	50	0.19	0.52	0.46	✗	0.17	0.45	0.59
	100	0.29	0.71	0.62	✗	0.21	0.65	0.76
	500	0.66	0.93	0.79	✗	0.35	0.85	0.94
K_d	50	0.36	0.62	0.47	0.43	0.38	0.51	0.68
	100	0.56	0.74	0.62	0.63	0.57	0.69	0.78
	500	0.82	0.94	0.75	0.87	0.79	0.85	0.94

Table 5.2 – Comparison of method AUPR values for the real dataset. For neighbourhood based averaging, results displayed correspond to $R = 16$ for both K values. For neighbourhood based smoothing, we displayed the results corresponding to $z = 8$ for both K values. Group based aggregation results correspond to the average aggregation function.

	Baseline	Preprocessing		Embedded		Postprocessing	
		Atlas	Neighb.	$\sum \Delta I$	Gp RF	Smooth.	Gp aggr.
$K = 1$	0.05	0.55	0.35	✗	0.04	0.09	0.69
K_d	0.08	0.59	0.49	0.14	0.04	0.39	0.99

Chapter 6

Group selection for the prognosis of Alzheimer's disease

Chapter overview

In this chapter, we propose a computer aided diagnosis system based on group selection. In particular, we decide to study more deeply the use of group importance score instead of feature ones, idea briefly discussed in Chapter 5. These scores help to rank properly groups of features and then to make a selection according to a ranking. A new forest can thus be fitted on the reduced learning set. As tree based ensemble methods are embedded feature selection algorithms for $K > 1$, their learning will make again a sort of feature selection. By consequence, even if some irrelevant features are in a selected group, the learning phase after group selection should correct that. The group selection method and the evaluation protocol of the ensemble classifier are explained in Section 6.2. We then illustrate the behaviour of the methods with two distinct datasets in Section 6.3. Finally, we finish the chapter with a short discussion and the proposition of new research directions. This chapter is the result of the following publication: M. Wehenkel, C. Bastin, C. Phillips, and P. Geurts. Tree ensemble methods and parcelling to identify brain areas related to Alzheimer's disease. In Pattern Recognition in Neuroimaging (PRNI), 2017 International Workshop on, *pages 1–4. IEEE, 2017.*

6.1 Problem definition

In this chapter, we study the possibility of using tree-based ensemble techniques both to design a CAD system for the prognosis of Alzheimer's disease and to improve our understanding of this disease. AD is currently the neurodegenerative disorder most often encountered in aged population [Brookmeyer et al., 2007] and predicting the susceptibility of a subject to develop AD before its onsets is thus highly relevant. To this goal, we develop a new approach based on feature selection and tree-based ensemble methods that can exploit a prior division of voxels into non overlapping brain regions (or groups) of interest. In particular, we propose to adapt a statistically interpretable measure defined in [Huynh-Thu et al., 2012] at the scale of groups of features, so as to select the most relevant brain regions on which to train the final classifier. Through experiments on our own dataset of 45 patients, we highlight the good behaviour of this approach and compare it with a linear SVM and standard (i.e., without group selection) tree-based methods, in terms of both predictive performance and interpretability.

We also carry out additional experiments on the OASIS dataset, a larger open-access dataset about dementia.

6.2 Computer aided prognosis system

In this section, we describe how is performed selection of groups of features. The main principle is to replace importance scores by statistically interpretable values as initially done in [Huynh-Thu et al., 2012] for biomarker feature selection. Once we have such statistics, the thresholding is not arbitrary anymore. Indeed, in statistics, the α threshold corresponds to the Type I error rate. More precisely, the percentage of error by rejecting the null hypothesis while it is true is at most α.

The originality of our method is to mimic this procedure at the group level. As we have seen in previous chapters, feature relevance scores are hardly reliable if the number of trees which has been used was not sufficient. Group importance score is an alternative post-processing method that can help to reduce this effect. Moreover, making feature reduction by groups will really help to remove totally useless information in the dataset. A successive and more accurate feature selection will thus be performed by the embedded feature selection procedure in tree ensemble algorithms.

6.2.1 Group selection method

Assuming that some prior, biologically plausible, division of voxels into G non-overlapping groups is provided by an expert (e.g., as defined in the AAL atlas [Tzourio-Mazoyer et al., 2002]), we propose the following two-step procedure to improve tree-based ensemble methods: a first ensemble is grown using all features as inputs and feature importance scores derived from this ensemble are exploited to select a small subset of the most relevant groups. Then, a new ensemble is grown using as inputs only the features from the most relevant groups and this latter ensemble is used as the final classification model.

To identify the most relevant groups and automatically select their number, we adapt at the group level the CER procedure proposed in [Huynh-Thu et al., 2012]. More precisely, group importance scores are computed by *averaging* the individual importance scores of their constituting voxels. Averaging is preferred over summation to avoid any bias due to differences in group cardinality. Groups are then ranked according to their importance scores. Let us denote by g_i the ith group in this ranking. A p-value like score is then associated to each group g_i of the ranking by estimating the probability

$$M(g_i) = P(rank(g_i) \leq i | H_R^{1 \rightarrow i-1}, H_I^{i \rightarrow G}),$$

where $H_R^{1 \rightarrow i-1}$ is the hypothesis that groups g_1 to g_{i-1} are relevant and $H_I^{i \rightarrow G}$ is the hypothesis that group g_i and all the groups ranked below g_i are irrelevant. The number of relevant groups is then computed as the maximum rank r for which $M(g_r) < \alpha$, with a small value α (fixed to 0.05 here). Following [Huynh-Thu et al., 2012], $M(g_i)$ scores are estimated by retraining tree ensembles on randomly permuted data (with 1000 repetitions): $H_R^{1 \rightarrow i-1}$ and $H_I^{i \rightarrow G}$ are simulated by keeping the class labels and the features in groups g_1 to g_{i-1} unchanged and by randomly permuting the features in groups g_i to g_G (using in this case the same permutation vector for all features so as to remain as close as possible to the original data distribution).

Intuitively, the idea behind this score is that a group which is really relevant should not be as well or better ranked than it is in the original data once we broke the link between the features in this group (and in all groups that follow in the original order)

and the output through the randomization procedure.

We apply this group selection procedure (denoted GS) using the AAL atlas with two distinct tree ensemble methods, Random Forests (RF) and Extremely Randomized Trees (ET), both implemented in MATLAB, and we use Breiman's mean decrease of Gini impurity measure to compute feature importance scores [Breiman, 2001], as in all the thesis. ET and RF both depend on two parameters: the number of trees T, fixed to 500, and the number K of features randomly picked at each node, set to its default value, i.e., the square root of the total number of input features. We did not attempt to optimise these parameters as tree-based ensemble methods are known to work well with default parameter setting and a sufficiently high number of trees (typically $\gg 100$). In particular, this number of trees is insufficient to directly interpret feature importance scores given the ratio $\frac{n}{m}$ we face to. However, group selection enables to solve this issue.

6.2.2 Validation protocol

Datasets

Two datasets are used to study the method we proposed here. Although they already have been described in details in Chapter 3, we briefly remind below their main characteristics. We perform experiments on :

- the CRC dataset, to classify MCI patients between stable MCI or future converters to AD. It is composed of 45 FDG-PET images from MCI patients. Four years later, patients have been clinically assessed and we found 22 AD against 23 stable MCIs.

- the OASIS dataset, to distinguish AD and CN individuals. There are one hundred images, one structural MRI for each person. This classification task is in general much more easy than the previous one. Nevertheless, the AD patients studied here are from mild to very mild AD, which makes the task more difficult. MRI images encode brain atrophy.

Performance Assessment

Accuracy, sensitivity, and specificity are evaluated by ten repeated 10-fold cross validation (as argued in [Varoquaux et al., 2016]), called runs here under.

We also compute *Receiver Operating Curves* (ROC) over the ten runs and areas under the curve (AUC). More precisely, this type of curves shows the evolution of the $sensitivity$ as a function of $1 - specificity$, i.e. the false positive rates.

With these cross validation procedures we evaluate the performance of the tree ensemble classifier composed of a group selection phase followed by a learning phase. The added value of group selection is highlighted by making the comparison with the tree algorithm performance without any feature selection. We are not able to evaluate independently group selection as we have achieved it in previous chapter as we do not know the right relevant groups in non artificial datasets.

Therefore, for comparison, we evaluate ET and RF without group selection with the same default setting and one linear method, SVM with a linear kernel, as implemented in PRoNTo [Schrouff et al., 2013b]. To compare all methods with default setting, the SVM model is first learnt with the parameter C set to its default value ($C = 1$), which was notably argued to be a reasonable choice in [Varoquaux et al., 2016]. To be as complete as possible, we also report below the performance obtained with SVM when C is

Table 6.1 – Method comparison on our own dataset. GS abbreviation is used for group selection. The asterisk (*) means parameter optimization.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
SVM	69.11 ± 3.54	66.82 ± 4.82	71.30 ± 4.67	73.97 ± 2.79
SVM*	67.33 ± 5.55	67.27 ± 4.69	67.39 ± 10.70	73.58 ± 5.85
RF	71.56 ± 4.29	63.64 ± 5.25	79.13 ± 4.49	77.79 ± 3.45
ET	74.67 ± 4.34	68.18 ± 8.57	80.87 ± 3.04	78.29 ± 3.27
GS/RF	76.44 ± 3.51	72.73 ± 4.79	80.00 ± 4.20	81.46 ± 2.27
GS/ET	79.11 ± 3.81	73.18 ± 3.98	84.78 ± 4.70	82.79 ± 3.10

optimised in a nested 10-fold cross-validation loop ($C = 10^{-3:1:3}$). For SVM, features are furthermore mean centred and normalized by their standard deviations before training. Such normalization is unnecessary with tree methods.

It is worth to note that group selection and SVM tuning are both performed only using the training folds to ensure unbiased estimates. We also carry out a significance paired t-test based on the ten repeated 10-fold cross validation estimates (with a 95% confidence level).

6.3 Results

6.3.1 CRC dataset

Method Performance

Table 6.1 and Figure 6.1 report performance results and ROC curves, respectively, for each method. SVM and RF exhibit comparable accuracy (i.e., no significant difference). GS improves the performance of both ET and RF according to all metrics and, despite the small size of the dataset, the differences are significant. ET, alone or with GS, performs significantly better than RF and GS/ET has the best area under the ROC curve (AUC). The corresponding ROC curve in Figure 6.1 shows that its sensitivity will be very poor if we want to keep a very low false positive rate. In practice, this is however not a serious drawback, since for medical prognosis, we are more concerned about reaching high sensitivities, even at the expense of the false positive rate (not to miss patients that will develop the disease).

Figure 6.2 illustrates the importances of the 10 most relevant groups, as well as the importances of the features within each of these groups, computed on the whole learning sample. All these groups have a $M(g)$ score lower than 0.05. Given the very small size of the dataset, trees are composed of very few tests and as a consequence, only a small number of features receive a non-zero importance score. As implied by the group importance measure (i.e., the average importance of the features in the group), the selected groups are such that most of their features are important, irrespectively of their size.

Method interpretability

SVM, as implemented in PRoNTo, provides interpretable results through weight maps per region [Schrouff et al., 2013a]. We compare in Table 6.2 areas identified with this method against those of tree methods, obtained by averaging voxel scores over folds and runs and subsequently aggregating these means per region. In this table, the highest

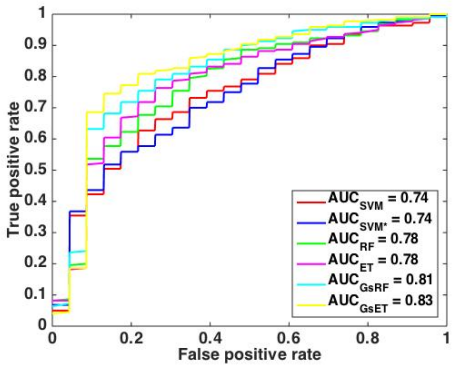


Figure 6.1 – ROC curves averaged over the ten runs for each method.

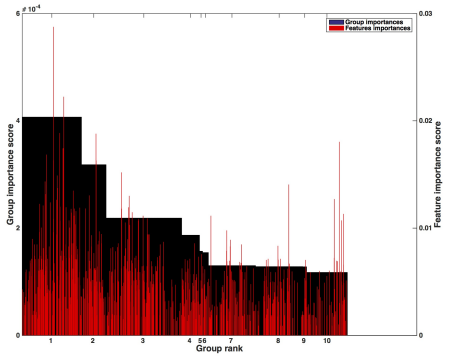


Figure 6.2 – Feature and group importances obtained with ET on our whole LS composed of PET data.

Table 6.2 – The five most contributing AAL regions for each method on our own dataset. L, resp. R, stands for left, resp. right, hemisphere.

Rank	SVM	SVM*	RF	ET	GS/RF	GS/ET
1	Parietal inf R	Parietal inf R	Angular R	Temporal mid R	Angular R	Temporal mid R
2	Angular R	Angular R	Temporal mid R	Angular R	Temporal mid R	Angular R
3	Vermis 8	Cerebelum 7b R	Parietal inf R	Temporal mid L	Parietal inf R	Temporal mid L
4	Cerebelum 7b R	Temporal mid R	Temporal mid L	Parietal inf R	Temporal mid L	Parietal inf R
5	Temporal mid R	Paracentral lobule L	Vermis 7	Vermis 7	Cuneus L	Temporal inf R

rank corresponds thus to the brain region of highest score. Obviously, first regions identified with SVM with or without parameter optimization are very similar. There are several regions common to RF, ET, GS/RF and GS/ET, e.g., the middle temporal gyrus (right and left hemispheres) and the angular gyrus. These regions are coherent with previous studies showing that MCI patients who are about to develop Alzheimer's disease exhibit more hypometabolic temporoparietal areas than MCI patients remaining stable in the next few years [Chetelat et al., 2003].

As an alternative to the importance-based ranking in Table 6.2, our group selection approach also enables the analysis of the most frequently selected (over the folds and runs) areas. This can give additional insights about the most relevant regions. We found that the regions the most frequently selected (i.e., more than half of the time) are the inferior parietal lobule (R), the angular gyrus (R) and the inferior (R) and middle (R and L) temporal gyrus, both for GS with ET or RF. More than just a region, group selection makes the tree algorithm focus only on a smaller number of features included in the most relevant areas and thus it identifies more precisely sub-regions interesting for the diagnosis limited to these important areas of the AAL atlas (cf. Fig. 6.3 for instance).

Relevance of the Atlas choice

Our results so far show that selecting the most relevant groups from the AAL atlas significantly improves the accuracy of tree-based ensemble methods (with respect to

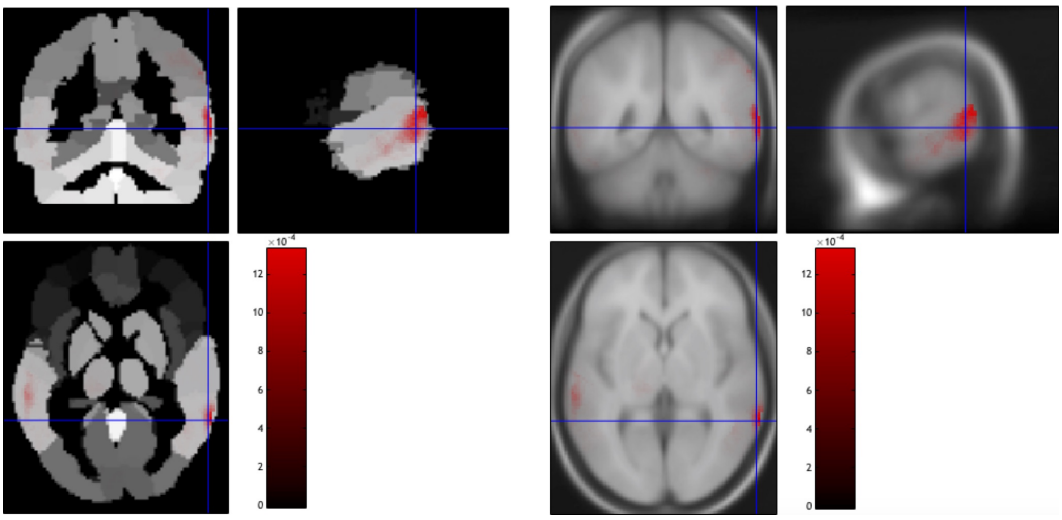


Figure 6.3 – Average of importance scores over folds and runs obtained with GS/ET inside the AAL atlas (left) and the *avg305T1.nii* SPM template (right).

Table 6.3 – Experiments with ten randomized atlases.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
GS/RF	70.00 ± 1.09	62.91 ± 1.38	76.78 ± 1.68	74.75 ± 0.74
GS/ET	71.60 ± 1.16	66.09 ± 1.83	76.87 ± 1.31	73.97 ± 0.70

no selection). In this section, as a sanity check, we would like to test if the AAL atlas is really a relevant choice of prior knowledge or if working with any randomly chosen groups of voxels would provide the same kind of improvement. To answer this question, we rerun our experiments using this time a random atlas, with the same group distribution (i.e. number and sizes of groups) as the AAL atlas but where the features are randomly assigned to the different groups. The experiment was repeated ten times with ten distinct randomized atlases and average results are reported in Table 6.3. These results are significantly worse than the results obtained by GS/RF and GS/ET in Table 6.1 (but also by RF and ET), which highlights the importance and relevance of the Atlas choice.

6.3.2 OASIS dataset

Method Performance

Table 6.4 reports the performance of each method on the OASIS database, using exactly the same protocol as in Table 6.1. All methods are this time very close to each other in terms of accuracy. SVM is not improved by parameter optimization and it slightly (but not significantly) outperforms tree-based methods. Group selection does not significantly improve (or deteriorate) the performance of RF and ET. We believe that these results could be indicative of the fact that dementia has a more global and distributed effect all over the grey matter in the brain, which makes the selection of a small subset of groups less useful than on the previous problem.

Table 6.4 – Method comparison on the OASIS dataset. GS abbreviation is used for group selection. The asterisk (*) means parameter optimization.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
SVM	67.90 ± 0.99	67.40 ± 0.97	68.40 ± 1.58	75.16 ± 0.82
SVM*	68.10 ± 1.85	67.00 ± 2.36	69.20 ± 3.68	74.50 ± 1.53
RF	64.90 ± 1.85	57.80 ± 2.57	72.00 ± 2.67	71.11 ± 1.35
ET	65.60 ± 1.58	59.80 ± 1.75	71.40 ± 2.32	72.61 ± 0.89
GS/RF	64.50 ± 1.78	58.40 ± 1.84	70.60 ± 2.99	68.97 ± 2.01
GS/ET	66.40 ± 2.84	59.20 ± 1.93	73.60 ± 4.88	70.32 ± 1.41

Table 6.5 – The five most contributing AAL regions for each method on the OASIS dataset. L, resp. R, stands for left, resp. right, hemisphere.

Rank	SVM	SVM*	RF	ET	GS/RF	GS/ET
1	Thalamus L	Cerebelum 10 R	Hippo-campus R	Hippo-campus R	Thalamus L	Hippo-campus R
2	Cerebelum 10 R	Cerebelum crus 2 R	Amygdala R	Amygdala R	Hippo-campus L	Amygdala R
3	Hippo-campus R	Cerebelum crus 2 L	Hippo-campus L	Hippo-campus L	Amygdala R	Hippo-campus L
4	Cerebelum crus 2 L	Paracentral lobule L	Amygdala L	Amygdala L	Amygdala L	Amygdala L
5	Cerebelum crus 2 R	Frontal sup L	ParaHippo-campal R	ParaHippo-campal R	Hippo-campus R	ParaHippo-campal R

Method Interpretability

Table 6.5 displays the top five regions picked with SVM weights and feature importance scores. Most of the regions of highest contributions for SVM are in the cerebellum. This result is not really in accordance with previous literature about distinguishing demented patients from control individuals. Indeed, the expected brain regions are notably those related to the hippocampus according to [Gosche et al., 2002, Klöppel et al., 2008]. RF and ET, with or without GS, seem to identify better the most important areas and the five first regions are the same for RF, ET and GS/ET whereas for SVM, the optimization causes the loss of the hippocampus in the five most important brain areas. Finally, GS/ET leads to a selection rate larger than 50% for nine regions: the hippocampus (L and R), the parahippocampal gyrus (L and R), the amygdala (L and R), the inferior occipital gyrus (L), the thalamus (L) and the middle temporal gyrus (R). With GS/RF, ten regions are highlighted: the same nine plus the left middle temporal gyrus.

6.4 Discussion

We have shown that, at least for the data and problems considered here, group selection with tree-based ensemble methods is competitive in terms of performance and interpretability with a method such as SVM traditionally used in neuroimaging.

Moreover, with our small dataset, group selection significantly improves the performance with respect to tree-based ensemble methods used without selection. This approach also provides additional insight about the regions relevant to diagnose a MCI patient who is likely to develop Alzheimer's disease within four years or to distinguish demented and healthy individuals thanks to the selection frequency of brain areas. In addition, group selection allows the method to focus in the second stage on discovering

subregions only in the most relevant regions.

We choose here to aggregate voxel importances within groups using the average, which has the advantage of being unbiased with respect to the size of the groups. This aggregation makes the method focus on identifying groups containing mostly informative voxels, versus identifying groups that contain only a few very important voxels. Different biases could be introduced by exploiting alternative aggregation operators, e.g., by computing the importance of a group as the sum of the importance of its features. We adapted here the permutation scheme of Huynh-Thu et al. [2012] to decide on the most significant groups. Interestingly, this permutation scheme would not be feasible at the level of voxels for our application, given the very high dimensionality. It would be interesting to investigate and adapt at the group level other statistical metrics proposed in the literature for tree-based ensemble methods, e.g., in [Huynh-Thu et al., 2012, Paul and Dupont, 2015]. Such variations of the procedure proposed in this chapter will be explored in Chapter 7.

Chapter 7

Statistical interpretation of group importance scores

Chapter overview

In this chapter, we pursue our objective to show the advantage of working at the group level for neuroimaging analysis with tree-based algorithms. Unlike previous Chapter 6, we focus here on safe group selection procedure, i.e. selection without irrelevant set of features. The goal is to select reliable groups of features relative to a phenotype of interest. In this case, we focus on the prognosis of Alzheimer's disease with FDG-PET scans. To validate all selection methods, we also provide a detailed analysis achieved on artificial datasets. This chapter is related to the following publication: M. Wehenkel, A. Sutera, C. Bastin, P. Geurts, and C. Phillips. Random Forests based group importance scores and their statistical interpretation: application for Alzheimer's disease. Frontiers in Neuroscience*, 12:411, 2018b. doi: 10.3389/fnins.2018.00411.*

7.1 Problem definition

One of the most commonly used ML methods in neuroimaging is Support Vector Machines (SVM) [Hearst et al., 1998]. The success of this method in this domain is due to its competitive performance when the number of features is large in comparison with the number of samples. In addition, when exploited with linear kernels, SVM provide weights for each voxel enabling the visualisation of brain patterns linked to the diagnosis [Vemuri et al., 2008, Zhang et al., 2011]. Nevertheless, these methods typically use the whole set of voxels to compute a prediction and, so, it is difficult to threshold the weights and interpret them in terms of their role importance in the patient condition. Sparsity-enforcing linear methods, such as Lasso or Elastic-net [Tibshirani, 1996, Zou and Hastie, 2005], are alternative techniques that embed a more explicit feature selection mechanism through a L1-penalization of the weight vector. These methods have been used with some success to analyse neuro-imaging data [Carroll et al., 2009, Ryali et al., 2010, Casanova et al., 2011]. Tree-based ensemble methods, such as Random Forests or Extremely Randomized Trees [Breiman, 2001, Geurts et al., 2006], are also known for their good predictive performance in high-dimensional/small sample size settings and furthermore provide interpretable results through feature importance scores. Their non-parametric nature makes them an interesting alternative to linear methods. Although they have not been studied extensively in the neuroimaging community, there

is evidence in the literature of their potential in such applications [Kuncheva et al., 2010, Langs et al., 2011, Gray et al., 2013, Ganz et al., 2015, Wehenkel et al., 2017].

When it comes to highlight brain regions involved in the studied disease, the main benefit of the aforementioned ML methods is their multivariate and non-parametric (for trees) nature, which potentially allows them to detect complex patterns in the data. Unlike statistical tests however, which associate to each problem feature a (corrected) p-value, scores extracted from ML methods, such as SVM weights and RF feature importances, can not be interpreted as easily. This makes very difficult the determination of a score threshold to distinguish the truly relevant features from the irrelevant ones in the resulting multivariate rankings. To circumvent this issue, the predictive performance of a ML model trained on a subset of features is therefore often used as a proxy to evaluate the relevance of the features in this subset and can be used to guide the search for the truly relevant features. For example, the regularisation level, and thus the sparsity, of sparse linear models can be tuned using cross-validation. Recursive feature elimination [Guyon et al., 2002, Guyon and Elisseeff, 2003] is an efficient procedure to find an optimal subset of features from SVM. A first SVM model is used to ranked all features. The lowest ranked features are then removed, a new model is retrained to rank the remaining features, and the process is repeated until no features are left. The feature subset that minimises cross-validation error in the resulting nested sequence is returned as the final optimal feature subset. In the context of Random Forests , Ganz et al. [2015] have proposed instead to remove iteratively the top ranked features and stop when the performance obtained on the remaining features is not better than random. While efficient mainly as a way to improve predictive performance, these methods do not really provide interpretable scores and, since cross-validation error is only a proxy for feature relevance, there is still a risk with these methods to either miss features or to select irrelevant ones [Huynh-Thu et al., 2012].

An alternative approach, proposed by several authors [Ge et al., 2003, Mourão-Miranda et al., 2005, Klöppel et al., 2008, Altmann et al., 2010, Huynh-Thu et al., 2012], is to exploit permutation tests in order to replace ML based scores by p-values like scores that are more interpretable and can be more easily thresholded. The general idea of these methods is to try to estimate for each score value v either the proportion of irrelevant features among those that have obtained a score higher than v (false discovery rate, FDR) or the probability that an irrelevant feature can reach such a high score (family-wise error rate, FWER). These values are estimated by exploiting more or less sophisticated permutation schemes that simulate feature irrelevance by randomly shuffling the labels. In order not to overestimate FDR or FWER values, these permutation schemes have to take into account the dependence that inevitably exists between importance scores derived from multivariate ML methods. Huynh-Thu et al. [2012] provide an empirical comparison of several of these methods, notably applied on RF importance scores, in the context of microarray classification problems in bioinformatics.

While very good results can be obtained by applying ML methods on neuroimaging data, identifying relevant features among hundreds of thousands of voxels with permutation tests is expected to be very challenging both computationally and statistically (as the more features, the higher the estimated FDR or FWER, because of multiple testing issues). In addition, the interpretability of a selection or ranking at the level of voxels is questionable. Because of the high expected spatial correlation among voxels, it is very likely than neighbouring voxels will be exchangeable when it comes to predict the output class, which will lead to unreliable importance scores as derived from ML methods. To circumvent this problem, Schrouff et al. [2013a] proposed to average absolute SVM weights in each region defined in a pre-existing anatomical brain atlas. This procedure improves interpretability by providing a ranking of brain regions, instead of

individual voxels, according to their contribution to the prediction. In [Schrouff et al., 2018], the same authors propose to address the problem directly at the training stage with a Multiple Kernel Learning (MKL) approach. A kernel is built on each brain region defined by an atlas. Weights are then attributed to each region during the learning process, with the weights penalized using a L1-norm to enforce their sparsity. Several works have also proposed adaptations of sparse linear methods to take into account data structure. For example, Michel et al. [2010] proposed a hierarchical agglomerative clustering procedure using variance minimisation and connectivity constraints that is combined in [Jenatton et al., 2012] with a sparse hierarchical regularisation approach to fit linear models. In this approach, there are as many groups of features as there are nodes in the hierarchical tree and each group is composed of all the descendants of a node. Weights are then attributed to each group such that if one node is unselected, all its descendants will have a zero weight too.

Following these latter works with linear methods, we would like in this chapter to investigate the benefit of group-based, instead of voxel-based, analyses in the context of Random Forests applied on neuroimaging data. Our first main contribution is the adaptation of Random Forests variable importance scores to rank and select groups of variables in the context of neuroimaging data. Assuming a prior division of the voxels into non overlapping groups, corresponding to different brain regions, we first propose several aggregation procedures to derive group importances from individual voxel importances. We then adapt the best permutation tests identified in [Huynh-Thu et al., 2012] to turn the resulting group importances into more statistically interpretable scores. Experiments are carried out on artificial datasets to analyse the behaviour of these methods in a setting where relevant groups are perfectly known. Our second contribution is the application of these methods on our own dataset of 45 patients for the prognosis of Alzheimer's disease. We report on this dataset the main groups identified with our methods and discuss their relevance with respect to prior knowledge about the disease. The methods are applied either on groups derived from existing brain atlases from the literature or on groups identified in a data-driven manner using clustering techniques. In addition, we also study on this dataset the influence of the main Random Forests parameters on both predictive performance and stability of group importance scores, from which we derive general guidelines for practitioners.

7.2 Methods

In this work, we are targeting the selection of relevant regions of interest in the brain for the prognosis of Alzheimer's disease with Random Forests. We assume a supervised learning setting, where we have a learning sample $LS = (X, Y)$ composed of n brain images of m voxel intensities each collected in a matrix $X \in \mathbb{R}^{n \times m}$ and of the n corresponding prognosis collected in a binary vector $Y \in \{0, 1\}^n$ (e.g., with 0 coding for stable MCI and 1 coding for MCI future converter). Following common machine learning terminology, voxel intensities will be also referred to as the *features* in what follows. From the learning sample, the goal is both to train a classification model that would classify as well as possible future brain images and to highlight the brain regions that are the most associated with the prognosis.

After a reminder of some basics relative to Random Forests, we then describe and motivate the three aggregation functions that will be evaluated later for computing importances of groups of features and explain how these groups can be obtained. Finally, we propose adaptations at the group level of the best techniques highlighted in [Huynh-Thu et al., 2012] to turn group importance scores into more statistically interpretable measures.

7.2.1 Random Forests and single variable importances

Random Forests [Breiman, 2001] is a supervised learning method that builds an ensemble of T decision trees [Breiman et al., 1984], as already introduced in Chapter 2. Several methods have been proposed to derive feature importance scores from a forest. In this thesis, as already stated we use the mean decrease of impurity (MDI) importance with the impurity measured with Gini impurity [Breiman, 2001, Louppe et al., 2013].

For the reminder, we mathematically defined the importance score of a variable x_i in a forest and we denoted it $\mathcal{I}(x_i)$ in Section 2.3 of Chapter 2.

7.2.2 Group importances

Importance scores as computed in Section 2.3 will give a ranking of the hundreds of thousands of voxels that typically compose neuroimaging data. Interpreting such ranking is not easy and typically requires to map these voxels on brain maps to visually identify brain regions with a significant number of high importance voxels. Statistically, one can also expect importances at the level of voxels to be rather unreliable given the typically very small size of neuro-imaging datasets. We propose here to exploit voxel individual importances to associate instead importances to sets of voxels. To this end, and to remain as general as possible, we assume the prior knowledge of a partition of the full set of voxels into several disjoint sets, which we are interested in relating to the disease status of the patients. Ways to define such partition will be discussed in the next section. Following the terminology used in sparse linear models, we will refer to the sets of voxels in a partition as *groups*. Given individual voxel importances as computed by a Random Forests model, group importances can be derived in several ways. Denoting by $X_G = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{\#G}}\}$ the set of features in a given group X_G of $\#G$ voxels, we will investigate three aggregation functions to derive group importances, computing respectively the sum, the average, and the max of the importances of the features in the group:

$$\mathcal{I}_{\text{sum}}(X_G) = \sum_{j=1}^{\#G} \mathcal{I}(x_{i_j}), \quad \mathcal{I}_{\text{avg}}(X_G) = \frac{1}{\#G} \sum_{j=1}^{\#G} \mathcal{I}(x_{i_j}),$$

$$\mathcal{I}_{\text{max}}(X_G) = \max_{j=1,\ldots,\#G} \mathcal{I}(x_{i_j}).$$

Some justifications about these choices have already been stated in Chapter 5 (Section 5.6).

7.2.3 Group definition

Computing group importances requires the availability of a partition of the voxels into groups. In this work, we will only consider partitions into contiguous sets of voxels, with groups thus corresponding to non-overlapping brain regions. Such partition will be referred to as an *atlas*. Two kinds of atlases can be investigated: (1) atlases derived manually from prior knowledge of the brain structure, such as the automated anatomical labelling (AAL) atlas [Tzourio-Mazoyer et al., 2002], and (2) data-driven atlases derived automatically from the learning sample using clustering techniques [e.g., Thirion et al., 2014]. We will focus our analysis in the rest of the chapter on the first family of atlases, which leads to more interpretable results. Some experiments with data-driven atlases on the real dataset are nevertheless presented in the appendices.

7.2.4 Group selection methods

Typically, most groups will receive a non-zero importance from the Random Forests model. From an importance ranking, it is therefore difficult to distinguish the truly relevant groups from the irrelevant ones. In this section, we propose to adapt at the group level, several methods that have been proposed in the literature to transform ML based importance scores into more statistically interpretable measures similar to p-values. This will help determining a threshold in the ranking below which all groups can be considered as irrelevant.

Beyond an improvement of interpretability, applying these techniques to groups of features instead of individual features has several additional advantages. First, some of these methods are very computationally demanding, as they require for each score computation, and thus for each feature, to retrain Random Forests several times with randomly permuted features or labels. This makes the application of the most demanding methods impossible at the level of voxels. Working at the group level, on the other hand, will reduce the number of scores to evaluate to a few hundreds only (depending on the size of the atlas) and therefore will strongly reduce computing times. Second, from a statistical point of view, one can expect aggregated group scores to be more stable than individual voxel scores. Combined with the strong reduction of the number of considered features, we expect that working at the group level will thus also improve the statistical power of the tests, which will lead to the selection of more significant brain regions than when dealing directly with voxels.

Huynh-Thu et al. [2012] have carried out an empirical comparison of several techniques to turn ML scores into statistical scores in the context of bioinformatics studies. We will present below the adaptation for groups of the three best methods identified in this study. Two of these methods, the *conditional error rate* (CER) and the *estimated false discovery rate* (eFDR), are based on models retrained on randomly permuted version of the original features, and one method, *mProbes*, train models with additional random features (called probes). mProbes and CER controls the family wise error rate and are recommended by Huynh-Thu et al. [2012] when a very low false positive rate is targeted (i.e. to minimise the number of groups selected that are not truly relevant), while the eFDR is comparatively less conservative as it controls the false discovery rate.

In our presentation of these methods, we assume that, from the learning sample LS, our machine learning algorithm has provided a score of importance s_i for each group, with $i = 1, \ldots, G$, using any aggregation function. Without loss of generality, groups are assumed to be ordered according to their importance score, such that g_i is the ith group in this ranking.

Multiple testing with random permutations

The goal of the CER and eFDR methods is to control the “family-wise error rate” (FWER) and the “false discovery rate" (FDR) respectively when choosing a threshold on the group importance scores. The FWER is the probability of selecting one or more false positives (irrelevant groups) among the groups that are identified as relevant, while the FDR is the expected rate of false positives among them [Storey and Tibshirani, 2003].

The *conditional error rate* method has been introduced in [Huynh-Thu et al., 2008] to overcome the limitations of the classic permutation-based FDR estimation techniques used for univariate statistical tests [Ge et al., 2003]. When applied to multivariate importance scores, these methods indeed usually overestimate the FDR, which leads to unreliable selections [Huynh-Thu et al., 2008]. The CER wants to estimate the prob-

ability to include an irrelevant group when selecting all groups until group g_i in the ranking. For group g_i, the conditional error rate is defined by:

$$CER_i = P(\max_{k=i,...,G} s_k^p \geq s_i \mid H_R^{1\rightarrow i-1}, H_I^{i\rightarrow G}), \tag{7.1}$$

where $H_R^{1\rightarrow i-1}$ is the hypothesis that groups g_1 to g_{i-1} are relevant, $H_I^{i\rightarrow G}$ is the hypothesis that group g_i and all the groups ranked below g_i are irrelevant and s_k^p is the importance score of the group k under these assumptions. The CER_i score for a given group g_i is estimated by retraining Random Forests on randomly permuted data (with P repetitions): class labels and features in groups g_1 to g_{i-1} are kept unchanged to simulate $H_R^{1\rightarrow i-1}$, while features in groups g_i to g_G are randomly permuted to simulate $H_I^{i\rightarrow G}$ (using the same permutation vector for all features so as to remain as close as possible to the original data distribution). The number of relevant groups is then computed as the maximum rank r for which CER_r is lower than a pre-defined risk α (with α typically set to 0.05).

In our previous work [Wehenkel et al., 2017], we proposed the following adaptation of the conditional error rate:

$$CER_i^r = P(rank(g_i) \leq i \mid H_R^{1\rightarrow i-1}, H_I^{i\rightarrow G}), \tag{7.2}$$

where the relevance score is replaced by the rank. The idea behind this score is that a group which is really relevant should not be as well or better ranked than it is in the original data once we break the link between the features in this group (and in all groups that follow in the original order) and the output through the randomization procedure. This adaptation is expected to be less restrictive than the CER in (7.1) and thus using the same α threshold, it should lead to a higher true positive rate at the expense however of the false positive rate.

Ge et al. [2008] propose to estimate the FDR with

$$eFDR_i = E\left[\frac{V_i}{V_i + i - 1}\middle| H_R^{1\rightarrow i-1}, H_I^{i\rightarrow G}\right], \tag{7.3}$$

where $H_R^{1\rightarrow i-1}$ and $H_I^{i\rightarrow G}$ are the same hypotheses as in (7.1) and V_i is the number of false positives. $eFDR_i$ is estimated in the following way. $H_R^{1\rightarrow i-1}$ and $H_I^{i\rightarrow G}$ are simulated using the same group-based permutation procedure as for the CER. V_i is computed, for each permutation, as:

$$V_i = \max_{k=1,...,G-i+1}\{k : s_{(1)}^p \geq s_i, s_{(2)}^p \geq s_{i+1}, ..., s_{(k)}^p \geq s_{i+k-1}\}, \tag{7.4}$$

with $s_{(k)}^p$ the kth largest value in $\{s_i^p, ..., s_G^p\}$ and s_k^p the relevance score of group g_k calculated from the randomly permuted data. V_i is thus the maximal number of randomly permuted groups, ordered according to their importance, whose importance exceeds the importance of the matching group ordered according to the original importance scores.

Utilization of random probes

A third method suggested by Huynh-Thu et al. [2012] is the mProbes approach, which is a variant of a method proposed in [Tuv et al., 2009]. When applied at the feature level, the idea of this method is to introduce as many random features as the input matrix contains originally, where each new random feature is generated by randomly permuting the values of one original feature. A Random Forests model is trained on the resulting dataset and is used to rank the features according to their importance. The experiment is repeated P times with new permutations and the FWER for a given

original feature is estimated by the proportion of the P runs where at least one random feature is better ranked than this feature.

The procedure can be easily adapted to groups. A random group is obtained from each original group by randomly shuffling the features within the group. Features within a group are permuted using the same permutation vector to keep feature correlations unchanged inside the groups. The FWER for a group g_i in the original ranking is then estimated by the proportion of Random Forests runs (among P) where at least one randomly permuted group is ranked better than group g_i.

This method is more efficient than CER and eFDR since it only requires to rerun Random Forests (with twice as much features however) P times, compared to $G \times P$ times with CER and eFDR, to get all group statistics.

7.3 Data and assessment protocol

7.3.1 Artificial datasets

In order to validate our methods in a situation where truly relevant features are already known, we generate artificial datasets for a linear classification problem. We used the same protocol of dataset generation as the one presented in Chapter 5 (Subsection 5.2), using $m = 500$ and $g = 50$ in all our experiments.

7.3.2 Real dataset

The real dataset used in this chapter is composed of 45 PET images and corresponds to the CRC dataset introduced in Subsection 3.3 of Chapter 3.

7.3.3 Atlas-based parcelling

For artificial datasets, the group structure is perfectly known in advance and it was used to define voxel groups. For real datasets, brain atlases are in general available for the sake of result interpretation. We thus decide to evaluate our methods with a prior division of the brain according to the brain structure as it is the simplest choice and the most interpretable one. In particular, the atlas we use is the AAL atlas [Tzourio-Mazoyer et al., 2002], composed of 116 distinct anatomical regions. The AAL atlas provides neuroanatomical labels only for gray matter areas. Our approach is thus by default limited to the gray matter. In addition, we provide in Appendix B results obtained with several data-driven atlases.

7.3.4 Group selection

Group importance scores are generated by Random Forests of 1000 trees by default, but larger values are also explored. Regarding the number of features randomly drawn at each split, i.e. the parameter K, we mainly explore two settings: $K = 1$ and $K = \sqrt{m}$. $K = \sqrt{m}$ is a common default setting which usually leads to good predictive performance on classification problems [Geurts, 2001]. $K = 1$ is an extreme setting, which amounts at selecting the feature for splitting a node fully at random. While this value of K is not expected to lead to optimal predictive performance, we tested this value for two reasons. First, it makes the tree construction very fast and independent of the total number of features. Second, it was shown in the theoretical analysis of [Louppe et al., 2013] to be the only setting that guarantees a fair treatment of all features by avoiding any masking effects between them. Indeed, when two features convey about the same

information about the output, using a value of $K > 1$ might prevent one of them to be selected at a given node when it is in competition with the other one. As a consequence, the importance of one of the two features will be greater than the importance of the other, while both features are almost equally important. Note however that using $K = 1$ is likely to lead to importance estimates of higher variance than using $K = \sqrt{m}$ and therefore to require building more trees for these estimates to reach convergence.

As in [Huynh-Thu et al., 2012], the permutation scheme for all statistical measures considers $P = 1000$ repetitions and the α threshold on all statistical scores is fixed to 0.05.

7.3.5 Performance metrics

Each method gives rise to a subset of relevant groups. In the case of artificial data, we are directly able to verify if this subset truly contains the right relevant groups. Method performance are thus evaluated in the case of artificial problems with the precision $\frac{TP}{S}$ and recall $\frac{TP}{P}$ with TP the number of truly relevant groups that have been selected, S the total number of selected groups and P the total number of truly relevant groups in the problem.

Independently of the use of a group selection method, it is interesting also to evaluate the quality of the group importance ranking. This ranking can be evaluated by computing the area under the precision-recall curve (AUPR), which plots the evolution of precision versus recall when selecting an increasing number of groups at the top of the ranking. The AUPR is equal to 1 when all truly relevant groups appear at the top of ranking and it is close to R/g, with g the number of groups, when groups are ranked randomly. To provide further comparison, we also evaluate the highest precision that can be achieved for a unitary recall and the highest recall that can be achieved for a unitary precision, respectively denoted *rec-1* and *prec-1* in the Results section. *rec-1* corresponds to the most conservative selection method that wants to avoid any false positive and *prec-1* corresponds to a method that does not want to miss any truly relevant feature. Note that these two methods are purely theoretical methods that can not be implemented in practice without a perfect knowledge of the relevant groups. Their performance is provided as baselines for comparison.

For the real dataset, as the truly relevant features (voxels or regions) are unknown, we can not evaluate performances through precision and recall as on the artificial datasets. As commonly done, we thus evaluate selection methods by comparing the regions found with the regions identified in the Alzheimer's disease literature. In addition, we also evaluate the different aggregation functions through the classification errors (estimated by cross-validation) of models trained using the most relevant groups found by each function. Finally, we further compare our methods with the MKL approach proposed in [Schrouff et al., 2018] using the AAL atlas. This method is close to ours in that it also performs feature selection at the level of regions. The C hyperparameters of this method is tuned using an internal ten-fold cross-validation loop (with C optimized in $10^{[-3:1:3]}$).

7.4 Results

We analyse in this section results obtained with artificial and real datasets.

7.4.1 Artificial datasets

Our goal in this section is to highlight the main properties of the group selection methods in a setting where relevant groups are known and one can thus assess quantitatively the capacity of the methods at selecting the correct groups.

Comparison of the aggregation functions

We first evaluate the quality of the group rankings obtained with the three aggregation functions: the *average*, the *sum*, and the *maximum*. AUPRs with the three functions are shown in Figures 7.1 and 7.2 respectively with $K = 1$ and $K = \sqrt{m}$, in both cases for an increasing number R of relevant groups and an increasing number of samples. All results are averaged over 20 randomly generated datasets.

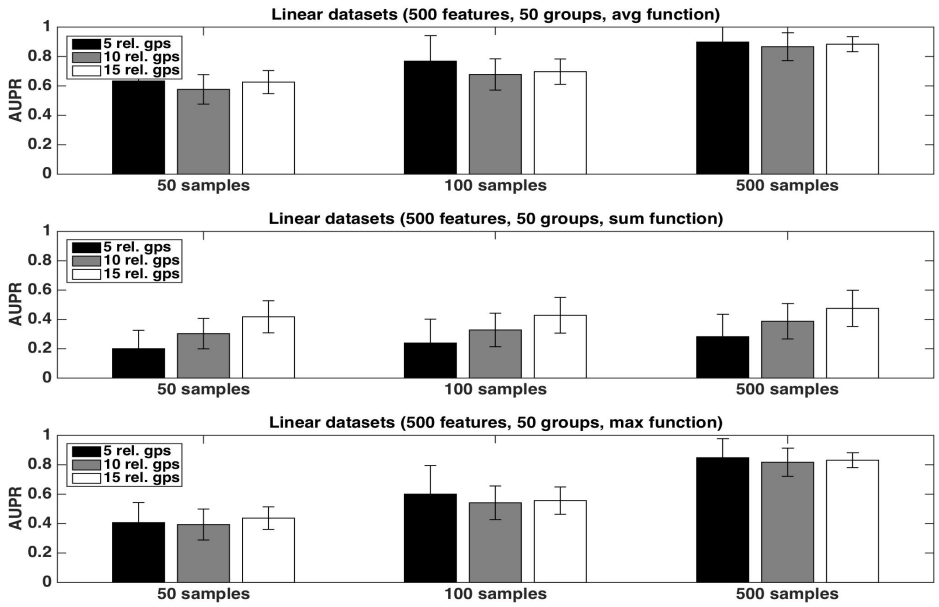


Figure 7.1 – Artificial datasets. AUPRs of Random Forests ($T = 1000$ and $K = 1$) ranking method with different aggregation functions, for different numbers of relevant groups and different sample sizes. Top is on *average* function, middle on *sum* function and bottom on *max* function. The AUPR values were averaged over 20 datasets in each case.

The *average* function is clearly producing the best rankings in all settings. The *max* function is competitive in large sample settings but it is clearly inferior with the smallest sample size. The *sum* is inferior to the two other functions in all settings, but its AUPRs are especially very bad when $K = 1$. We attribute the bad performance of the sum in this setting to its bias towards groups of large size. Indeed, when $K = 1$, features used to split are selected uniformly at random among all features and thus there are more splits based on features from larger groups in the trees. As a consequence, even if each feature of a large irrelevant group will receive a low importance, when summing them, the importances of their group might still be comparable with the importances of small relevant groups. As a confirmation of this effect, we indeed observe a strong correlation

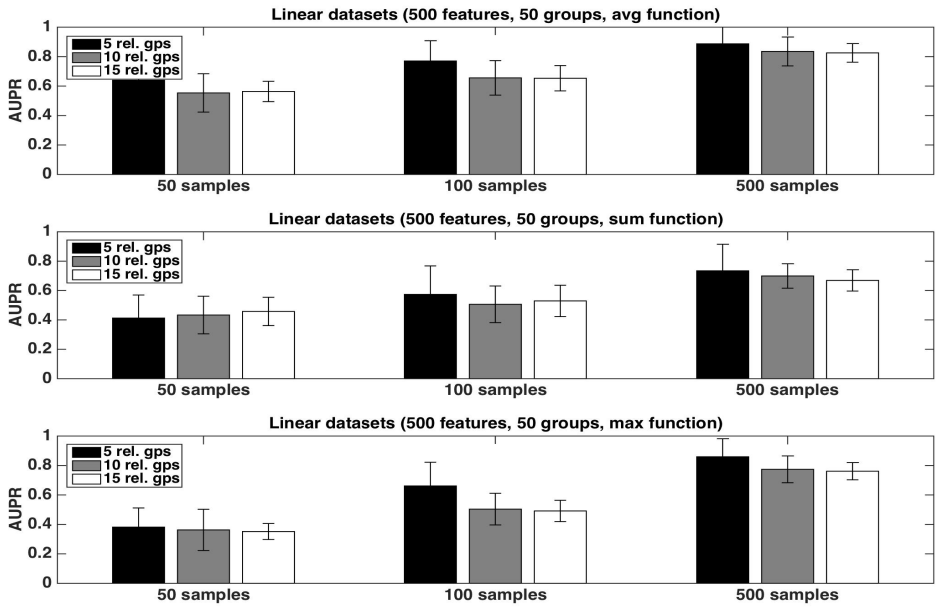


Figure 7.2 – Artificial datasets. AUPRs of Random Forests ($T = 1000$ and $K = \sqrt{m}$) ranking method with different aggregation functions, for different numbers of relevant groups and different sample sizes. Top is on *average* function, middle on *sum* function and bottom on *max* function. The AUPR values were averaged over 20 datasets in each case.

between group importances and group sizes when using the sum function. Although still present, the effect is reduced with $K = \sqrt{m}$, as in this case, features from irrelevant groups are put in competition with features from relevant groups and have thus less chance to be selected in the trees.

As expected, the AUPRs increase in all cases when the number of samples increases. Except for the *max* function, the AUPRs slightly decrease with the number of relevant groups.

Comparison of statistical scores

In Figure 7.3, we show, both for $K = 1$ and $K = \sqrt{m}$, how the different statistical group measures evolve with the rank for the three aggregation functions. In all cases, the group importances decrease rapidly and then much more slowly, suggesting that only a few groups contain most of the information. The only exception is the maximum group importance with $K = 1$, which decreases slowly from the beginning. Statistical scores mostly show the expected behaviours. CER and mProbes, which both estimate the FWER, have similar evolutions. The statistical measures they compute remain close to zero for 3 or 4 groups and then increase very abruptly towards 1. As expected, eFDR, which estimates the FDR, leads to a slower increase of its statistical score towards 1 also after 3 or 4 groups. CER^r has the slowest progression in all cases, except with the *sum* function and $K = \sqrt{m}$ where it increases more rapidly than the other scores. All statistical scores are directly close to 1 with the *sum* function when $K = 1$, showing that the ranking provided by this group importance does not behave well. Note that the point

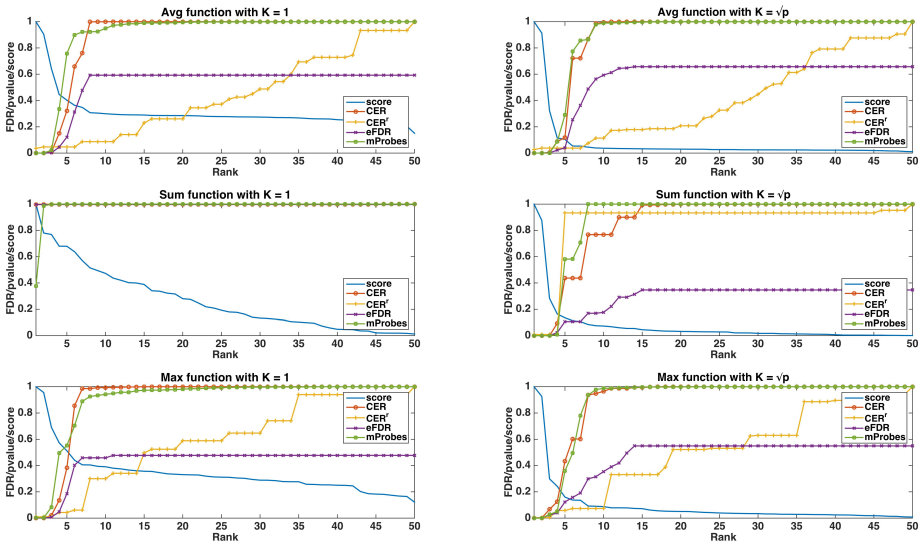


Figure 7.3 – Artificial datasets (500 features, 50 groups, 5 relevant groups, 100 samples). Curves of the importance scores ($T = 1000$, $K = 1$ and $K = \sqrt{m}$) and the different selection methods obtained on a linear dataset. Score in the legend stand for group importance score.

where most statistical scores start raising is consistent with the position in the ranking at which irrelevant groups starts appearing: with the *average*, the first irrelevant group is at the fifth position in the ranking, whatever K. With the *sum*, the fourth group is the first irrelevant one for both K. With the *max*, the first irrelevant group is the first one with $K = 1$ and the fifth one with $K = \sqrt{m}$.

Table 7.1 compares methods when they are used for feature or group selection directly. We report in this table the average (over 20 datasets) number of groups selected by all four methods, the average number of features that are contained in these groups, and the average number of relevant groups among the selected ones. As a comparison, we also provide in the same table, the number of features and (relevant) groups selected when the four statistical scores are computed at the level of features instead of groups. In this case, a group is considered as selected as soon as one of its feature is selected.

Several interesting observations can be made from this table. When working at the group level, the *average* aggregation leads to the highest number of selected groups with CER, mProbes, and eFDR. With the CER^r, more groups are found with the *max* aggregation. Except with the CER^r, it is interesting to note that working at the level of features instead of groups actually leads to the selection of less groups than using the average group importance. This supports our previous argument that working at the group level is actually beneficial in terms of statistical power. The CER and the mProbes methods seem to only find relevant groups since the average number of selected groups always exactly matches the number of selected relevant groups. For the eFDR, a few selected groups are actually irrelevant as these two numbers do not exactly match. The CER^r on the other hand seems to select much more irrelevant groups. In particular, its precision is very poor when it is used at the feature level. These results will be confirmed in the next section. Finally, for all methods, using $K = \sqrt{m}$ allows to find

Table 7.1 – Average number of features selected F ($\alpha = 0.05$) and number of corresponding groups G and relevant groups TG on linear artificial datasets (500 variables, 50 groups, 5 relevant groups and 100 samples) for each method. RF means Random Forests without any aggregation function. Bold text and underlined text are for best number of relevant groups over all aggregation functions and over all selection methods respectively.

		CER			CER^r			eFDR			mProbes		
		F	G	TG	F	G	TG	F	G	TG	F	G	TG
$K = 1$	RF	7.15	1.55	1.55	47.75	20.35	**3.70**	11.85	1.85	1.75	1.75	0.2	0.2
	avg	18.50	2.20	**2.20**	14.85	1.40	1.30	21.45	2.70	**2.60**	16.00	1.75	**1.75**
	$\sum$	5	0.30	0.30	7.75	0.45	0.45	7.5	0.40	0.40	7.40	0.35	0.35
	max	14.90	1.60	1.60	28.35	3.10	2.55	17.45	1.75	1.65	11	1.10	1.10
$K = \sqrt{m}$	RF	7.05	1.45	1.45	61.10	23.80	**3.80**	11.20	2.05	1.75	11.05	1.65	1.65
	avg	19.80	2.75	**2.75**	25.90	2.65	2.15	22.55	3.15	**3.05**	20.45	2.70	**2.70**
	$\sum$	16.35	1.40	1.40	23.55	2.20	2.15	17.35	1.55	1.55	22.75	2.00	2.00
	max	12.50	1.65	1.65	35.90	4.00	2.90	14.50	1.80	1.75	12.95	1.75	1.75

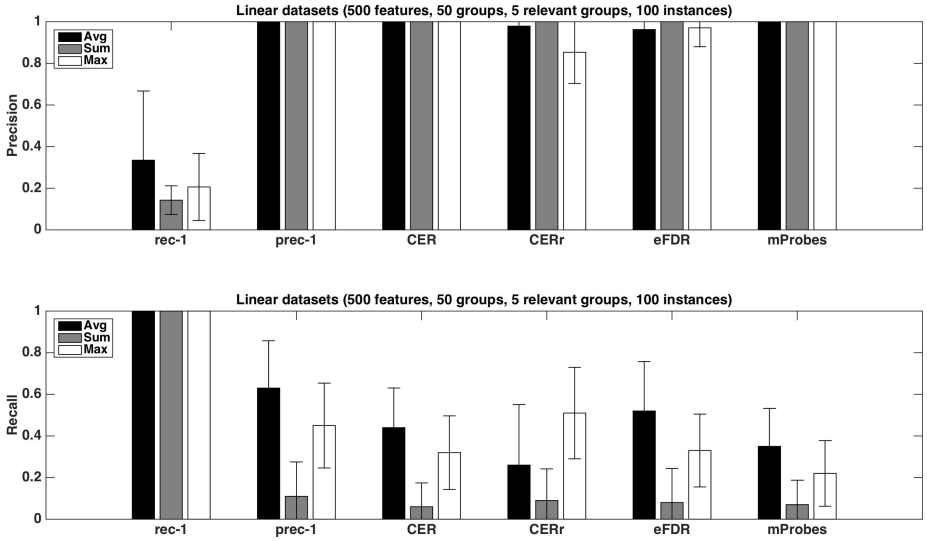


Figure 7.4 – Artificial datasets. Precision and recall of each selection method ($T = 1000$ and $K = 1$) for the three different aggregation functions investigated. We used a selection threshold $\alpha = 0.05$. The precision and recall values were averaged over 20 datasets in each case.

more (relevant) groups that using $K = 1$.

Precision and recall

Figure 7.4 shows the precision and recall of each method with the different aggregation functions averaged over 20 datasets, with $K = 1$. As already noticed from Table 7.1, the precision is close to one for all methods except the CER^r with *max*. None of the proposed methods can reach a recall equal or higher than the one of prec-1. Except for CER^r for which the recall is the highest when *max* is used, the other methods obtain the best results with the *average* aggregation function. eFDR with averaging obtains the highest recall among the proposed methods, while the recalls of mProbes and CER

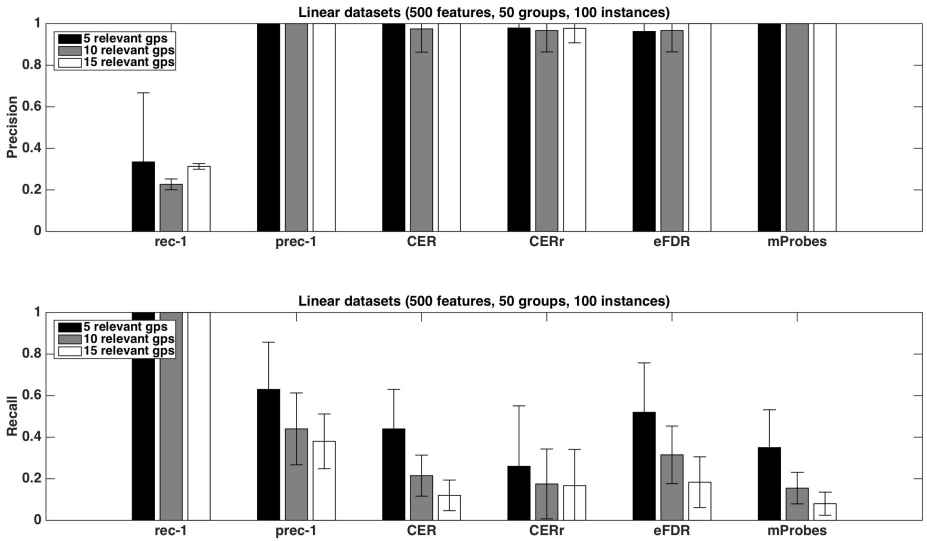


Figure 7.5 – Artificial datasets. Precision and recall of each selection method ($T = 1000$ and $K = 1$) for different numbers of relevant groups. We used the *average* function as ranking method and a selection threshold $\alpha = 0.05$. The precision and recall values were averaged over 20 datasets in each case.

are very close.

Figure 7.5 shows the impact of the number of relevant groups on precision and recall, with the *average* function. Precisions are mostly unaffected while recalls decrease when the number of relevant groups increases. Given that the recall is the proportion of relevant groups found by the methods, this suggests that the number of selected groups does not grow proportionally with the number of relevant groups.

Finally, as expected, increasing the number of samples in datasets helps to improve the performances. This phenomenon is illustrated in Figure 7.6. With 500 samples, recall of CER, eFDR and mProbes are getting closer to recall of prec-1. Unfortunately, such a ratio is in general not encountered in neuroimaging problem. Improvement of recall value is really less impressive for CER^r. This latter method also exhibits a lower precision than the other ones.

Summary

The comparison of the aggregation functions shows that the *average* and the *max* functions work better than the *sum* function, due to a bias of this latter aggregation function towards large groups, in particular when $K = 1$. The *average* function provides better AUPR scores than the *max* in small sample setting, while both methods are close with larger sample sizes. Concerning RF parameters, $K = \sqrt{m}$ is clearly a better choice than $K = 1$ as it enables to detect more relevant groups, at the expense however of computing times. Among statistical scores, CER and mProbes select no false positives while eFDR selects a few and CER^r a lot. Finally, our results show that working at the group level is beneficial because it allows to select more relevant groups than working at the level of individual features.

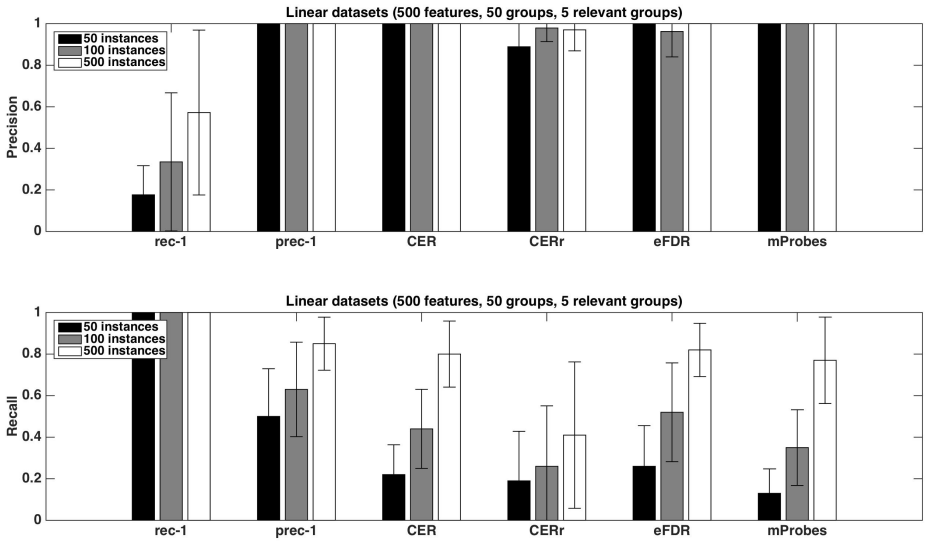


Figure 7.6 – Artificial datasets. Precision and recall of each selection method ($T = 1000$ and $K = 1$) for different numbers of sample sizes. We used the *average* function as ranking method and a selection threshold $\alpha = 0.05$. The precision and recall values were averaged over 20 datasets in each case.

7.4.2 Real dataset

In this section, we present results obtained with the group selection methods on the CRC dataset related to Alzheimer's prognosis. This dataset constitutes a very challenging problem for ML methods, as it contains a very large number of features (around $200,000$ voxels) and only few dozens of samples (45 patients). We will first study the predictive performance of Random Forests on this dataset (in comparison with the MKL method) and study the impact of its main parameters, T and K, on both error rates and group ranking. We will then investigate the behaviour of the group selection methods, depending on the aggregation function and Random Forests parameters. Finally, we will analyse the groups found by these methods in the light of prior knowledge about Alzheimer's prognosis.

Predictive performance and group ranking

Figure 7.7 shows the evolution of the error rate depending on parameters K and T. Errors in this figure are obtained as averaged over ten repeated ten fold cross-validation runs. The error rate for $T = 1000$ reaches its minimum value at around $K = 1000$ (which is close to $K = \sqrt{m}$). Moreover, the error decreases as the number of trees T composing the forest increases and stabilises at around $T = 1000$. With default parameters ($T = 1000$ and $K = \sqrt{m}$), Random Forests reach an error rate of 28.89%, which is much better than the error rate of a classifier always predicting the majority class (49%). This suggests that despite the small size of the dataset, Random Forests are able to extract meaningful information from the data.

While default values perform well in terms of error rate, it is interesting to study the impact of these parameters also on the stability of the group rankings. Using the AAL

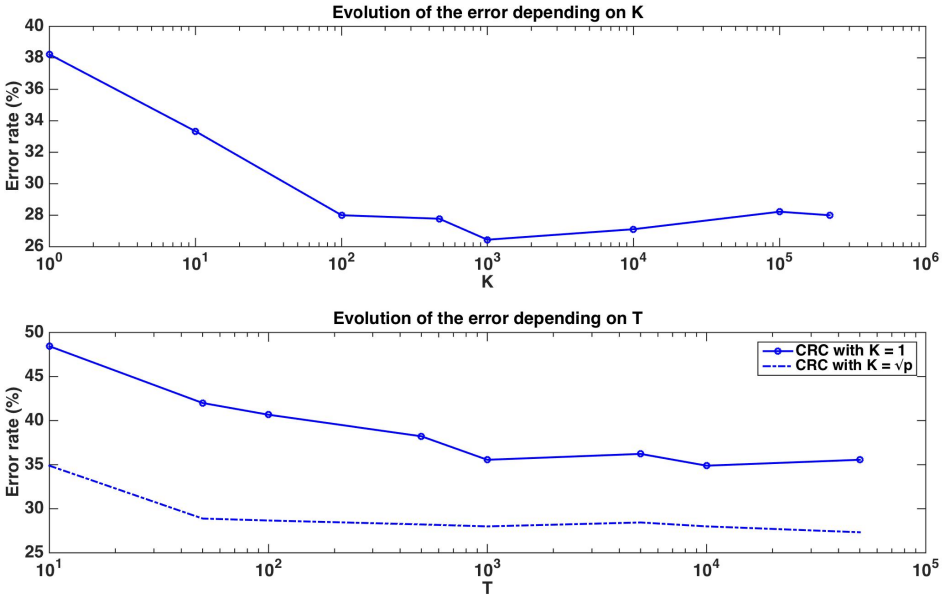
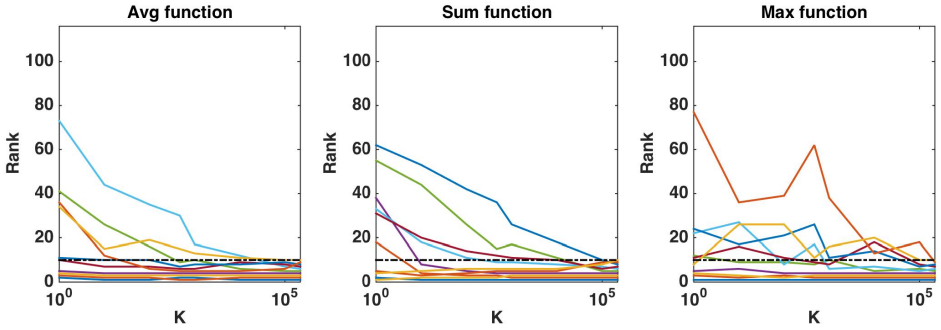


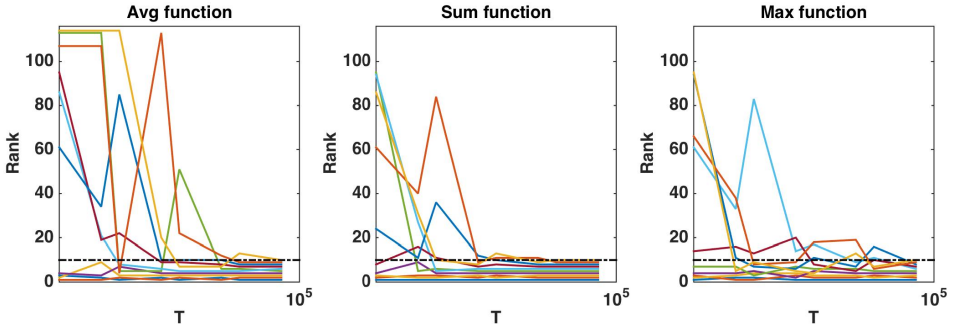
Figure 7.7 – Real dataset. Error rates of a Random Forests classifier as a function of K parameter value for $T = 1000$ (top figure) and as a function of the number of trees T for $K = 1$ or $K = \sqrt{m}$ (bottom figure). Errors are evaluated with a ten repeated ten fold cross validation procedure.

atlas, Figure 7.8(a) plots the evolution of the rank of ten groups when K is increased from 1 to m (and T is set to $10,000$), for the three aggregation functions. The ten groups are selected as the ten most important groups when $K = m$, so that their rank converges towards $\{1, 2, \ldots, 10\}$ when K grows to m. The top four groups seem to remain the same whatever the value of K, as soon as K is not too small. The evolution of the rank of the other groups is however more chaotic, whatever the aggregation function, and some groups only reach the top ten when K is very close to m. Figure 7.8(b) shows the effect of T on the ranking of the top ten groups obtained with $K = \sqrt{m}$ and $T = 10,000$. The number of trees has clearly a strong impact on rankings. Only the top 2 or 3 groups are already at their final position when T is small. The *sum* aggregation converges faster than the other two and it is the only one to have its top 10 groups fixed for $T < 10,000$. As already shown in [Huynh-Thu et al., 2012], this suggests that more trees are required to stabilise feature importances than to reach optimal predictive performance.

To compare and analyse further the different aggregation functions, Figure 7.9 shows the group importances and the individual voxel importances within each group for the top five groups ranked by the three aggregation functions (with $K = \sqrt{m}$ and $T = 10,000$). The first four groups found by all aggregation functions are the same, while each function highlights a different group at the fifth position. The order between the top four groups however differs between functions but these differences can be explained. For example, the *sum* function puts group 85, which is larger, in front group 66, while they are ordered inversely with the *max* and *average* that are less sensitive to group sizes. While the maximum importance in group 85 is higher than the maximum importance in group 62, the *average* function prefers group 62 over group 85 because group 62 has less voxels of small or zero importance proportionally to its size.



(a) Evolution of the rank as a function of K parameter value for the first ten regions obtained ($T = 10,000$ and $K = m$).



(b) Evolution of the rank as a function of T parameter value for the first ten regions obtained with $T = 10,000$ and $K = \sqrt{m}$.

Figure 7.8 – Real dataset. Importance scores are computed for the AAL atlas and for each aggregation function. Black horizontal dotted line represents the 10^{th} ranking position.

Without knowledge of the truly relevant groups, we can not assess group rankings using the AUPR, like we did on the artificial datasets. One common indirect way to evaluate a ranking is to build models using the top ranked features and see how it improves error rates: the better the ranking, the faster the error decreases when groups are introduced in the model. Figure 7.10 shows how the cross-validation error evolves when we progressively introduce the groups in the model following the rankings obtained with the three aggregation functions. The value 0 corresponds to a model always predicting the majority class without using any features. Errors were estimated as the average over five repeated ten-fold cross-validation runs. To avoid any selection bias in the evaluation, the groups are reranked at each iteration of each 10-fold cross-validation run without using the test fold. For comparison, we also show on the same plot the error obtained by Random Forests trained using all voxels (about 28%). One can see from this plot that it is possible to decrease the error rate from 28% (when using all voxels) to about 20% whatever the aggregation function used, suggesting that all group rankings contain informative groups at their top. This is consistent with results in Figure 7.9 that show that the top of the rankings are similar. The minimal error is reached in the three cases with a very small number of groups (respectively 8, 2, and 3 groups for the *average*, the *sum*, and the *max* aggregation), but the position of this minimum is clearly very unstable and almost optimal performance is reached with only a couple of groups. With the *max* and *average* aggregations (resp. with *sum* aggregation), the improvement over RF with all voxels is statistically significant (according to a t-test with risk level 0.05) when from 1 to 4 (resp. 5) groups are selected.

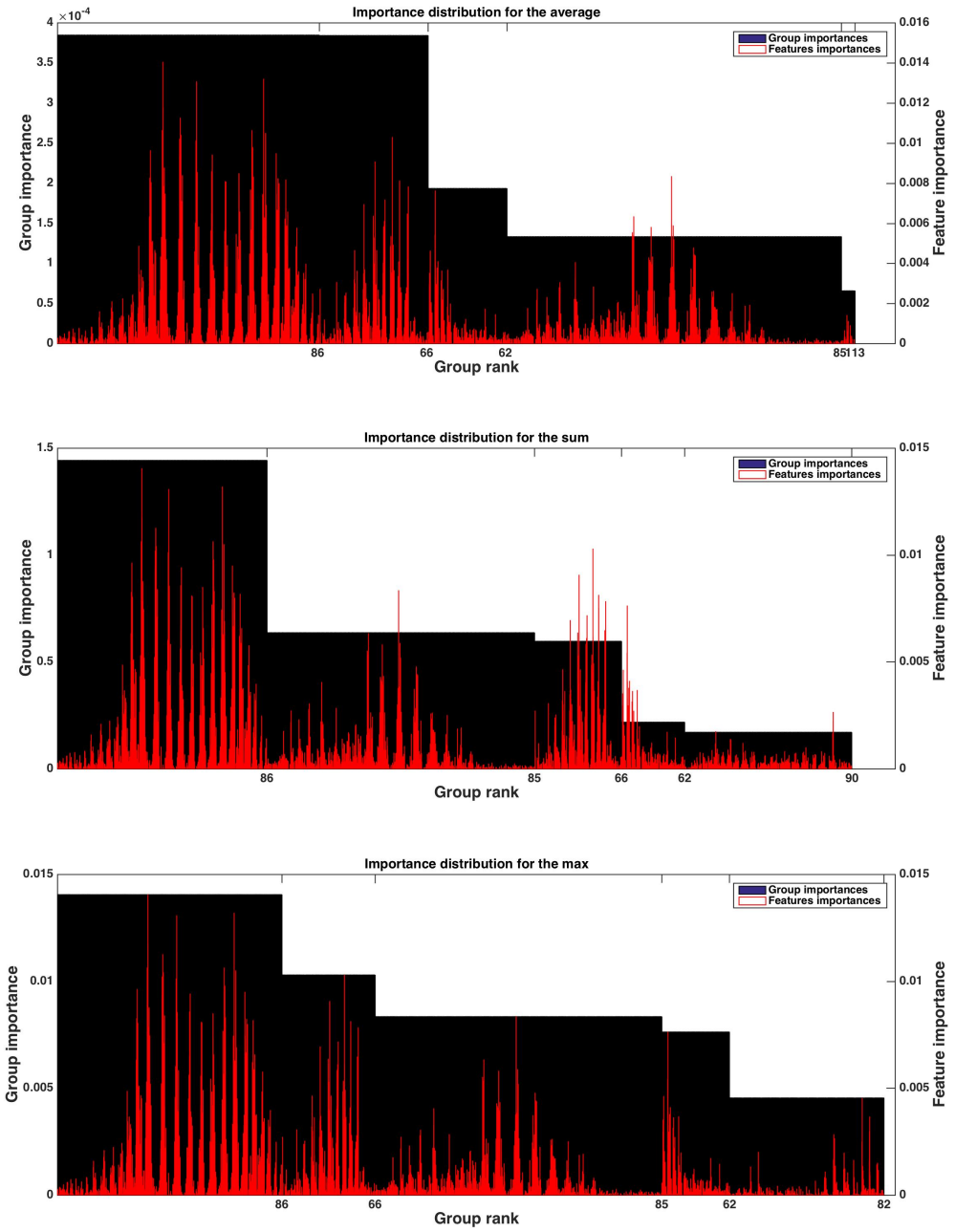


Figure 7.9 – Group and individual voxel importances for the five groups of highest ranks, from top to bottom when using the *average*, *sum*, and *max* aggregation functions (with $K = \sqrt{m}$ and $T = 10,000$). X-axis shows the group number at the position of the last voxel within the group. Note that left y-axis is group importance, while right y-axis is voxel importance (different scales have been used for readibility).

As a baseline for the obtained error rates, we also compare Random Forests with the MKL method proposed in [Schrouff et al., 2018] using the AAL atlas and setting

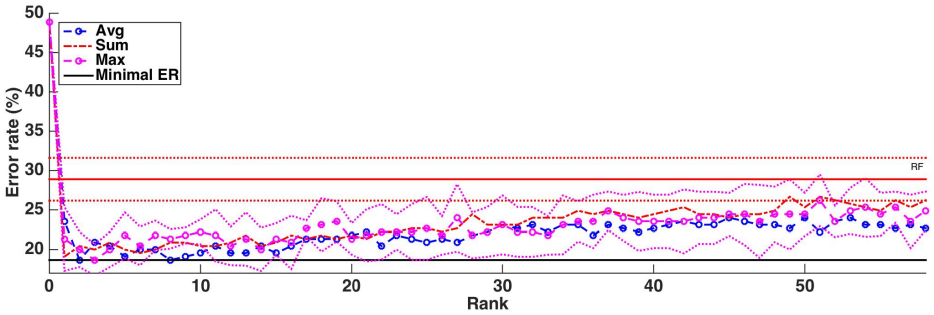


Figure 7.10 – Real dataset. Error rates of a Random Forests classifier as a function of the number of groups included in the model ($T = 1000$ and $K = \sqrt{m}$). Errors are evaluated with a five repeated ten fold cross validation procedure. Error rate obtained with Random Forests ($T = 1000$ and $K = \sqrt{m}$) is represented by a red horizontal line, while the minimum error rate is represented by the black horizontal line. Standard deviations for RF and the *average* function are represented as dotted lines.

its parameter with an internal cross-validation as explained in the method section. We obtain an error rate of 39.56% with MKL, which is worse than the 28.89% error rate obtained with Random Forests and default setting.

Group selection methods

We analyse here the output of the different group selection methods.

In Appendix B (Figure B.1), we illustrate how the statistical scores change when going down in the ranking, for each method and aggregation function. Scores of importance aggregated with the *sum* show a faster decrease than with the other aggregating functions. Regarding the selection methods, mProbes and CER are clearly more conservative methods since their statistical scores rapidly increase in all cases. The behaviour of CER^r is more dependent on the aggregation function used. With the *sum*, it is nearly as restrictive as mProbes and CER. However, when combined with *average* or *max*, score evolution is much more progressive, even more than eFDR. These observations are consistent with results on the artificial problems.

Table 7.2 summarizes the number of groups selected by each method (with $\alpha = 0.05$) with every aggregation functions and different RF parameter settings. Overall, we observe very sparse results, with only a few, if any, groups selected in most settings. This is not surprising given the small size of the dataset and observations in the previous section (that show that an optimal error rate can be achieved with only a couple of groups). The only exception is the CER^r method which selects more groups with the *average* and *max* aggregation. We know however from experiments on the artificial data that this method has a low precision. In general, the *max* and *average* aggregation functions lead to the selection of more groups than the *sum*. Overall, with $K = 1$, increasing the number of trees from 1000 to $10,000$ increases the number of selected groups. With $K = \sqrt{m}$, increasing T does not seem to affect the number of selected groups however. Comparing $K = 1$ and $T = 10,000$ with $K = \sqrt{m}$ and $T = 1000$, we see that the latter setting leads to more groups overall, in particular when the mProbes method is used (it does not select any group with the *average* and *max* aggregation when $K = 1$). This suggests to set $K = \sqrt{m}$ and $T \geq 1000$ to maximize the number of groups selected. Note

Table 7.2 – Number of regions selected ($\alpha = 0.05$) for the real dataset for each method depending on the aggregation function.

$(K;T)$	CER			CER^r			eFDR			mProbes		
	avg	$\sum$	max	*avg*	$\sum$	max	*avg*	$\sum$	max	*avg*	$\sum$	max
$(1;1000)$	0	2	1	9	0	2	0	2	1	0	1	0
$(1;10,000)$	0	2	3	10	0	8	0	2	3	0	2	0
$(\sqrt{m};1000)$	2	3	2	17	1	8	0	3	3	2	3	1
$(\sqrt{m};10,000)$	0	3	2	>4	1	>4	0	4	4	2	5	3

however that this advise should be taken with caution since K could also affect the proportion of false positives among the selected groups.

Interpretability

In this section, we analyse more precisely the groups selected with our methods and discuss them in the light of existing literature about MCI prognosis.

Several studies have looked at brain regions that impact AD prognosis. In univariate studies about AD prodromal stages, differences between MCI converters and non-converters have been identified to be localized mainly in the right temporoparietal and in the medial frontal area [Chetelat et al., 2003, Chételat et al., 2005, Drzezga et al., 2003, Nielsen et al., 2017]. More precisely, according to the regions defined by the AAL atlas, the regions that are the most often identified as relevant for AD conversion are the superior temporal, the inferior parietal and the superior medial frontal. Several publications have also highlighted the middle temporal gyrus (right and left hemispheres) and the right angular gyrus [Morbelli et al., 2010]. There thus only exist few regions discriminating converters and non-converters. Moreover, it remains a difficult task to differentiate these two classes of MCI as observed differences are generally very subtle. We believe this is consistent with the fact that most group selection methods only can find few regions.

It remains to be checked whether the regions found belong to the ones mentioned in the literature. For this purpose, we list in Table 7.3 the first ten top-ranked regions for all aggregation functions and for all RF parameter settings. With the *average* aggregation, brain regions at the first five positions vary a lot depending on the parameters T and K. Rankings are more stable with the *sum* and *max* aggregation functions. Overall, regions highlighted as the most important by all of these rankings are mostly consistent with studies about MCI progression towards Alzheimer's disease.

Table 7.3 can also be analysed along with the lines corresponding to the AAL atlas in Table 7.2 that show how many groups are considered as relevant by each selection method. To illustrate such analysis, we report in Table 7.4 for the top ranked AAL regions with the three aggregation functions the statistical scores estimated by CER, eFDR, and mProbes (with $K = \sqrt{m}$ and $T = 10,000$). In each column, we only report the statistical scores until the first score higher than $\alpha = 0.05$ (as next groups will be considered irrelevant anyway). We also provide a visual representation of this table in the brain space in Figure 7.11. Two groups are systematically selected as relevant (except by CER and eFDR with the *average* aggregation). These are the angular gyrus (right) and the middle temporal gyrus (right). With the *sum* and the *max* aggregations, eFDR and mProbes both selects two additional regions: the middle temporal gyrus (left) and the inferior parietal (right). Finally, only mProbes selects the inferior temporal gyrus

Table 7.3 – Real dataset. First ten regions of rankings provided by Random Forests with different aggregation functions depending on parameters K and T. R and L stand for right and left hemispheres respectively, g. for gyrus, c. for cortex, sup. for superior and inf. for inferior, $\triangle$ denotes triangular part of the inferior frontal gyrus.

	$(K;T)=(1;1000)$	$(K;T)=(1;10,000)$	$(K;T)=(\sqrt{m};1000)$	$(K;T)=(\sqrt{m};10,000)$
avg	Cuneus c. (L)	Angular g. (R)	Angular g. (R)	Middle temporal g. (R)
	Angular g. (R)	Middle temporal g. (R)	Middle temporal g. (R)	Angular g. (R)
	Middle temporal g. (R)	Vermic lob. 8	Inf. parietal (R)	Inf. parietal (R)
	Inf. parietal (R)	Vermic lob. 7	Middle temporal g. (L)	Middle temporal g. (L)
	Cerebelum 7b (R)	Middle temporal g. (L)	Thalamus (L)	Vermic lob. 7
	Inf. temporal g. (R)	Inf. parietal (R)	Cuneus c. (L)	Inf. temporal g. (R)
	Middle temporal g. (L)	Vermic lob. 6	Vermic lob. 8	Cuneus c. (L)
	Inf. temporal g. (L)	Inf. temporal g. (R)	Sup. temporal g. (R)	Inf. temporal g. (L)
	Sup. occipital g. (L)	Cuneus c. (L)	Heschl (R)	Sup. temporal g. (R)
	Olfactory (L)	Inf. temporal g. (L)	Inf. temporal g. (R)	Vermic lob. 8
$\sum$	Middle temporal g. (L)	Middle temporal g. (L)	Middle temporal g. (R)	Middle temporal g. (R)
	Middle temporal g. (R)	Middle temporal g. (R)	Middle temporal g. (L)	Middle temporal g. (L)
	Inf. temporal g. (R)	Middle frontal g. (L)	Angular g. (R)	Angular g. (R)
	Inf. temporal g. (L)	Inf. temporal g. (R)	Inf. parietal (R)	Inf. parietal (R)
	Middle frontal g. (L)	Inf. temporal g. (L)	Inf. temporal g. (R)	Inf. temporal g. (R)
	Middle occipital g. (L)	Middle frontal g. (R)	Sup. temporal g. (R)	Inf. temporal g. (L)
	Precuneus (R)	Middle occipital g. (L)	Inf. temporal g. (L)	Sup. temporal g. (R)
	Middle frontal g. (R)	Sup. frontal g. (L)	Sup. temporal g. (L)	Cuneus c. (L)
	Cuneus c. (L)	PreCuneus c. (L)	Cuneus c. (L)	Sup. temporal g. (L)
	Sup. frontal g. (R)	Sup. temporal g. (R)	Cerebelum 6 (L)	Cerebelum 6 (R)
max	Middle temporal g. (R)	Middle temporal g. (L)	Middle temporal g. (R)	Middle temporal g. (R)
	Calcarine (R)	Sup. temporal g. (R)	Middle temporal g. (L)	Angular g. (R)
	Middle temporal g. (L)	Middle temporal g. (R)	Angular g. (R)	Middle temporal g. (L)
	Inf. temporal g. (R)	Inf. temporal g. (R)	Sup. temporal g. (R)	Inf. parietal (R)
	Angular g. (R)	Inf. temporal g. (L)	Inf. parietal (R)	Sup. temporal g. (R)
	Cuneus c. (L)	Angular g. (R)	PreCuneus c. (L)	Inf. temporal g. (R)
	Inf. parietal (L)	Hippocampus (R)	Calcarine (L)	Cerebelum 8 (L)
	Inf. frontal g. $\triangle$ (L)	Thalamus (L)	Cuneus c. (L)	Cerebelum 6 (L)
	Inf. temporal g. (L)	Calcarine (L)	Inf. temporal g. (R)	Middle occipital g. (R)
	Postcentral g. (R)	Inf. occipital g. (L)	Temporal pole (Mid. temp. g. L)	Thalamus (L)

Table 7.4 – Real dataset. First top-ranked regions and corresponding statistical scores for different aggregation functions with $K=\sqrt{m}$ and $T=10,000$. R and L stand for right and left hemisphere respectively, g. for gyrus, sup. for superior and inf. for inferior.

	Regions	CER	eFDR	mProbes
avg	Middle temporal g. (R)	0.057	0.057	0.046
	Angular g. (R)			0.042
	Inf. parietal (R)			0.215
$\sum$	Middle temporal g. (R)	0	0	0.001
	Middle temporal g. (L)	0.006	0.003	0.013
	Angular g. (R)	0.006	0.003	0.020
	Inf. parietal (R)	0.081	0.030	0.042
	Inf. temporal g. (R)		0.051	0.046
	Inf. temporal g. (L)			0.065
max	Middle temporal g. (R)	0.010	0.010	0.003
	Angular g. (R)	0.028	0.016	0.019
	Middle temporal g. (L)	0.060	0.023	0.049
	Inf. parietal (R)		0.026	0.206
	Sup. temporal g. (R)		0.136	

(right) with the *max* aggregation. These five regions are very consistent with the regions highlighted in the literature, as regions related to parietal and temporal areas are those that came out the most frequently.

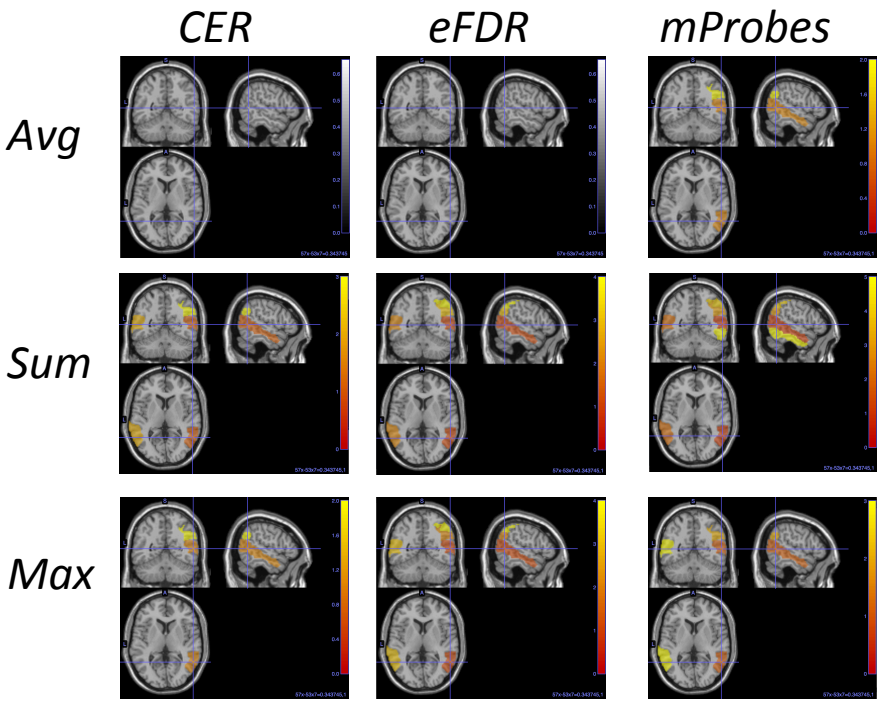


Figure 7.11 – AAL regions selected with each method and each aggregation function for $K = \sqrt{m}$ and $T = 10,000$. This picture is a visual representation of Table 7.4. The blob color provides information about the ranking: the more red the region is the better is its rank.

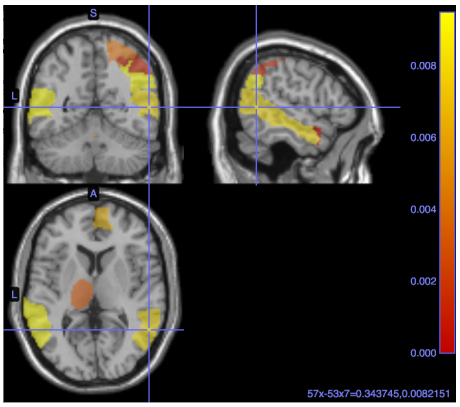


Figure 7.12 – AAL regions selected with MKL. Weights are averaged over the ten repeated ten folds. The blob color provides information about the ranking: the more red the region is, the lower is its weight.

In comparison, averaging weights obtained over folds with MKL highlights the following regions in its top ten (in decreasing order of the weights): the middle temporal

gyrus (right), the angular gyrus (right), the vermis 6 lobule, the thalamus (left), the frontal superior medial gyrus (right), the middle temporal gyrus (left), the vermis 8 lobule, the cerebelum 10 (left), the superior parietal gyrus (right) and the hippocampus (right). Regions selected are visually represented in the brain space in Figure 7.12. Although there are actually 76 regions over 116 with a non zero weight, we can however analyse how these weights are distributed. The first ranked region has a weight of 30 while the nine others show a weight between 9 an 2. After the tenth region, weights are slowly decreasing towards zero. The MKL top ten has three regions (out of five) in common with those highlighted with group selection methods, with two at the top of its ranking. Differences between the two lists are not unexpected given the different natures of the models (linear versus non-parametric) and would deserve to be analysed more thoroughly.

7.5 Discussion

We proposed several methods based on Random Forests to select relevant groups of features on the basis of interpretable statistical scores. These methods are helpful in neuroimaging to improve the interpretability with respect to standard ML based analysis carried out at the level of voxels. In addition to an improvement of interpretability, group selection methods potentially exhibit a higher statistical power than feature selection methods. We have confirmed this through experiments on artificial datasets, where group methods are able to detect more relevant groups than similar methods working at the level of features. Moreover, on high dimensional datasets, computing statistical scores at the level of features can rapidly become very computational demanding. Working at the level of groups has thus only advantages when such groups naturally exist in the data.

We first assessed the behaviour of the different group selection methods through experiments on artificial problems where a group structure is imposed. By design, CER and mProbes are more conservative than eFDR and CER^r. In terms of interpretability, CER^r is less reliable because it selects in general too many groups that can include a significant number of false positives. The other methods appear to be safe overall as they do not wrongly declare irrelevant groups as relevant. The comparison of the different aggregation functions to derive group importances from feature importances has shown that the *average* provides the best results, followed by the *max* and then the *sum*. The *sum* should be used carefully with $K = 1$ when groups of very different sizes are present in the data. Interestingly, when combined with group selection methods, this problem can however be diagnosed without knowledge of the truly relevant groups, as it will lead to no group being selected as relevant by any group selection method. Concerning the Random Forests parameters, $K = \sqrt{m}$ appears to detect more relevant groups than $K = 1$, although this latter setting has been shown theoretically to not suffer from masking effects.

We then applied the methods on the CRC dataset related to Alzheimer's Disease prognosis. The conclusions are almost the same on this dataset, when methods are compared in terms of the number of groups they select. CER and mProbes are more conservative than eFDR and CER^r. We thus recommend to use CER and mProbes to have more confidence in the selected regions. If reducing computing times is important, mProbes is clearly the best choice among these two as it only requires one round of permutations. Note however that all methods can be easily parallelized and in general, we believe that computing times should not really be an issue, especially when working with groups. As on the artificial datasets, using $K = \sqrt{m}$ leads to more groups than $K = 1$, as does increasing the number of trees T, which should be taken larger than for

optimizing error rate alone. No strong conclusion can be drawn concerning the aggregation functions however. Taking the sum does not show the same pathological behaviour as on the artificial data and actually can lead to more selected groups (e.g., in Table 7.4).

Concerning Alzheimer's Disease prognosis, results are encouraging although they deserve to be analysed more thoroughly. Error rates are acceptable in our opinion, especially taking into account the small size of the dataset. They can be furthermore reduced significantly by focusing on a couple of groups. The group selection methods have highlighted several regions, e.g., the middle temporal gyrus (right) and the angular gyrus (right), that are consistent with the literature on MCI progression towards AD.

As future work, we would like to confirm our results on additional real datasets. While we focus here on interpretability, we would like also to explore more the possibility to improve predictive performance through group selection. Figure 7.10 shows that selecting a few groups can lead to improved error rates and in [Wehenkel et al., 2017], we showed that building Random Forests on the top of groups selected by CER^r could also improve performance. In our work, we use groups only to post-process Random Forests importance scores, but did not change anything in the way forests are grown. It would be interesting to investigate ways to incorporate groups directly during the Random Forests training stage, as it is done for example in the MKL framework [Schrouff et al., 2018] or in sparse linear methods [Jenatton et al., 2012].

Part III

Applications to Alzheimer's disease

Chapter 8

Transfer learning for the characterization of Alzheimer's disease

Chapter overview

The subject of this chapter is the inference of unknown information from one dataset to another one in order to characterize two different metabolic profiles of AD patients. A first dataset is composed of imaging data from a few AD patients labelled in two classes depending on their FDG-PET scan. Neuropsychological assessments would be necessary in order to characterise clinically these two profiles. However, such information is missing. The second database composed of AD patients, which is larger, includes such clinical information. The former database will be used to label the latter one depending on the metabolic profile. The second database will provide the clinical information we are looking for. Part of the results presented in this chapter are in process to be submitted in a journal as "F. Meyer, M. Wehenkel, C. Phillips, P. Geurts, R. Hustinx, C. Bernard, C. Bastin, E. Salmon. Characterization of a temporoparietal junction subtype of Alzheimer's disease.".

8.1 Problem definition

FDG-PET imaging allows the measurement of brain energetic metabolism and its changes induced, for example, by the progression of Alzheimer's disease (AD) [Herholz et al., 2002, Mosconi, 2005, Mosconi et al., 2009]. The temporo-pariental lobes, parts of the frontal cortex, and the posterior cingulate gyrus of AD patients are typically suffering of hypometabolism [Hoffman et al., 2000, Minoshima et al., 1994, 1997]. The largest deficits due to the disease mainly concern the cingulate and temporoparietal cortices [Bohnen et al., 2012].

During a study carried out at the Cyclotron Research Centre (CRC) on fifty-two AD patients, neurologists observed that the FDG-PET scan of about half of the patients displayed some differences from the typical expected hypometabolic profile of an AD patient. Nevertheless all these patients exhibited the same clinical profile leading to a probable AD dementia diagnostic. The dataset is thus composed of two categories of PET images: a typical AD hypometabolic profile (labelled AD_t) and an atypical hypometabolic profile in the temporo-parietal junction (labelled AD_{TPJ}).

These brain differences could be the cause of cognitive deficits. Therefore clinical and neuropsychological patient information from both classes could exhibit some differences. Unfortunately complete neuropsychological assessment had not been performed for all these patients when they were PET-scanned. A more complete dataset, including both FDG-PET scans and a more complete neuropsychological assessment, could be used to differentiate typical (AD_t) and atypical (AD_{TPJ}) patients. These data are extracted from the ADNI database (see Chapter 3). The ADNI database includes a large number of neuropsychological scores per subject and many more subjects than the CRC dataset. This will allow a more detailed analysis of the atypical TPJ patients. A key assumption here is that ADNI includes both types, typical and TPJ, of AD patients even though such differences have not yet been discussed in previous studies, as far as we know. Since AD patients in the CRC dataset were selected using the same clinical diagnosis as for ADNI, such assumption seems realistic. However the AD patients in ADNI have not been classified into sub-groups, according to their FDG profile: typical, TPJ, or even other types. Therefore we suggest to proceed with a *transfer learning* procedure.

We thus proceed in two steps in this chapter. Firstly we train a classifier with the CRC database in order to predict the labels of the ADNI database. In a second step, we analyse the clinical information provided by the ADNI database.

8.2 Data

Data used in this chapter have been shortly introduced in Chapter 3 (Section 3.3). Demographic details are provided in Table 3.1. We provide here supplementary details necessary for a full understanding of the problem.

8.2.1 CRC_2 data

All 52 AD subjects enrolled in this study met the clinical criteria defined by the National Institute on Aging-Alzheimer's Association for probable AD dementia [McKhann et al., 2011]. At the Cyclotron Research Centre, researchers visually detected hypometabolism differences in patient FDG PET scans and confirmed their observations by a statistical univariate study performed with SPM12. Results are detailed in "F. Meyer. *Description et étude d'un nouveau profil d'atteinte métabolique au PET-FDG chez des patients atteints de démence d'Alzheimer probable.* Unpublished master thesis, University of Liège, Liège, Belgium, 2017.", available in the institutional repository ORBI.

They compared AD_t patients with controls and AD_{TPJ} patients with controls and observed significant differences in both comparisons. Their statistical analysis highlighted an hypometalism in the precuenus (bilaterally) and in the posterior cingular cortex. For TPJ patients, results mainly showed an hypometabolism in the left side of the precuneus and in the temporoparietal junction. Compared to the TPJ patients, the typical ones, showed significantly reduced metabolism in the precuneus (left), the posterior cingulate cortex (left) and right inferior lateral temporal cortex. Conversely the TPJ patients had a significantly higher hypometabolism near the left TPJ.

8.2.2 ADNI data

In the ADNI database, 207 AD patients were pre-selected for their FDG-PET scan recorded at their study entrance and the availability of many neuropsychological assessments for all of them. During the course of the study, these data were visually labelled by a medical expert into the typical and TPJ subgroups in order to validate our

approach.

Neuroscience experts from the CRC a priori selected 33 scores of interest among the whole set of clinical scores provided by the ADNI database. The scores of interest are listed in Table 8.1. More importantly it shows if the score is available for all patients or only for some of the subjects. Most scores are discrete (only the RAVLTp score is continuous) and some scores span a much larger scale than others. A full description of the scores is provided here below:

- ADAS scores [Rosen et al., 1984] from Question 1 to Question 12 (ADAS_Q1 to ADAS_Q12);

  ADAS (for Alzheimer's Disease Assessment Scale) test consists on a series of questions evaluating cognitive functions, mood and behaviour. The different questions address different cognitive domains (language, praxis, visuospatial abilities, short term and long term memory,...). It can help to determine the stage of the disease in which the person is and to follow the evolution of the disease.

- NPI score [Cummings et al., 1994];

  NPI, for Neuropsychiatric Inventory, test estimates several behavioural disturbances occurring in different types of dementia.

- CDR score [Morris, 1993];

  The Clinical Dementia Rating score evaluates the severity of dementia on a "5-point" scale (0, 0.5, 1, 2 , 3 from normal to severe dementia). The domains that are estimated to obtain this score are memory, orientation, judgment and problem solving, community affairs, home and hobbies, and personal care.

- MMSE score [Folstein et al., 1975, Tombaugh and McIntyre, 1992];

  The Mini Mental State Examination is carried out to evaluate the level of cognitive impairments of a patient on a scale from 0 to 30. It is composed of questions of seven different domains (e.g. orientation to time or to place, language,...).

- RAVLT scores [Rey, 1941, Schmidt et al., 1996];

  The Rey Auditory Verbal Learning Test has been conceived to assess verbal learning and episodic memory. We take interest on five sub-scores of this test (RAVLTi for immediate, RAVLTl for learning, RAVLTf for forgetting, RAVLTp for perc forgetting, RAVLTd for delayed).

- FAQ score [Pfeffer et al., 1982];

  The Functional Assessment Questionnaire evaluates the level of assistance a person needs in daily living tasks.

- Clock Drawing scores [Kaplan, 1983];

  The Clock Drawing test evaluates the capacity of the subject to draw a clock in response to verbal instructions and to make a visual copy of a clock. This test thus corresponds to two different scores (ClockD for drawing and ClockC for copying).

- Logical memory score - Immediate and delayed recall (LMIR and LMDR) [Wechsler, 1987];

  Such test estimates the capacity of a subject to recall a short story read to the subject immediately or with a delay.

- Digit span forward and backward scores (DSFw and DSBkw) [Wechsler, 1987];

  This test measures the memory working in forward and backward directions, i.e. sequences of numbers have to be recalled and repeated in both directions.

- Category fluency scores for animals and vegetables (FluA and FluV);

  It measures flexible retrieval of information in semantic memory, by assessing the ability to report as many examples as possible for a given category.

- Trail making scores (TMTA and TMTB) [Reitan, 1958];

  Two scores are of interest because this test is typically composed of two trials, evaluating notably speed of processing tasks (visuo-motor speed) and executive functioning. These scores consist in time duration, in seconds.

- Digit symbol substitution score (Digit) [Wechsler and De Lemos, 1981];

  With this score, we can also measure the processing speed and short-term memory.

- Boston naming test score (BNT) [Kaplan et al., 2001].

  Such score are notably useful to detect possible deficits in object recognition and access to lexico-semantic knowledge.

8.3 Methods

Transfer learning is the ability to apply knowledge learnt from an original task to another target task presenting similarities with the first one [Arnold et al., 2007, Pan and Yang, 2010]. In particular, we take interest on *transductive* transfer learning in this work, i.e. the former and the latter problems have both the same classification tasks whereas input domains are similar but not identical, and labels are unavailable in target domain. More precisely, the CRC_2 dataset will be used to learn brain differences between AD_{TPJ} and AD_t. This information will then be used to predict ADNI labels and then to extract clinical scores of interest to pursue the characterization of typical and TPJ AD patients.

8.3.1 ADNI labelling

In this subsection, we explain the methods used in order to predict ADNI labels from the CRC_2 database.

Group selection

In order to possibly reduce the feature set, we first run a ten repeated ten fold cross validation procedure with a combination of a CER_r group selection procedure (500 Extra-trees, $K = \sqrt{m}$, $\alpha = 0.05$) and an atlas division according to the AAL atlas [Tzourio-Mazoyer et al., 2002]. Among the one hundred folds, groups having been selected by the CER_r method more than half of the time are considered as relevant.

These groups are used to reduce the input matrix and fit a final classifier on CRC_2 dataset (a forest of 500 Extra-trees with $K = \sqrt{m}$). ADNI labels are therefore predicted using this classifier. We called these labels Y_{ET}.

Classification evaluation

The ten repeated cross validation procedure performed above provides an estimate of the CRC_2 classifier performance.

In a second stage, the accuracy of machine learning labelling is estimated by a "reverse learning" approach. A new Extra-trees model is learnt from ADNI data with Y_{ET}

Table 8.1 – Clinical score information: type of score (continuous vs. discrete), number of distinct values in the dataset ($\#val$), total number of missing values ($\#MV$) and number of missing values for TPJ labels ($\#_{TPJ}MV$) for each clinical score. TPJ labels used in this table are the ones attributed by a medical expert.

Clinical score	Continuous	Discrete	$\#val$	$\#MV$	$\#_{TPJ}MV$
ADAS_Q1		✓	23	0	0
ADAS_Q2		✓	5	0	0
ADAS_Q3		✓	5	0	0
ADAS_Q4		✓	8	0	0
ADAS_Q5		✓	9	0	0
ADAS_Q6		✓	5	0	0
ADAS_Q7		✓	9	0	0
ADAS_Q8		✓	25	0	0
ADAS_Q9		✓	6	0	0
ADAS_Q10		✓	5	0	0
ADAS_Q11		✓	5	0	0
ADAS_Q12		✓	5	0	0
NPI		✓	17	13	4
CDR		✓	16	3	0
MMSE		✓	10	3	0
RAVLTi		✓	39	3	0
RAVLTl		✓	10	4	0
RAVLTf		✓	11	4	0
RAVLTp	✓		19	5	0
RAVLTd		✓	9	0	0
FAQ		✓	29	3	0
ClockD		✓	6	0	0
ClockC		✓	6	0	0
LMIR		✓	13	0	0
LMDR		✓	9	0	0
DSFw		✓	10	125	28
DSBkw		✓	11	125	28
FluA		✓	25	0	0
FluV		✓	15	125	28
TMTA		✓	85	2	0
TMTB		✓	108	18	3
Digit		✓	41	125	28
BNT		✓	27	1	1

labels. It is then tested on CRC_2 data. During the course of the study, the ADNI dataset was visually labelled by a medical expert (these labels are denoted Y_M). To compare these labels with those obtained with machine learning, we build a classifier using Y_M labels in order to predict CRC_2 labels. Moreover, we also directly compare Y_{ET} and Y_M labels.

All classifiers are compared using area under ROC curves (sensitivity in function of $1 -$ specificity) averaged over 50 runs. In each case, Y_{ET} labels using group selection are compared to the ones obtained without group selection. We also analyse in the next subsection Y_{ET} labels obtained without using the group selection approach.

8.3.2 Clinical scores detection

The final aim of the study is to detect clinical scores potentially explaining the two distinct classes of AD patients. To pursue this objective, we learn a classifier using the clinical scores as input data and the ADNI labels as output data. An ensemble of trees fitted on this database will provide importance scores and thus an intuition about the most relevant clinical scores to distinguish AD_t and AD_{TPJ} patients. Experiments are performed using Y_{ET} and Y_M labels to compare results with both labelling approaches.

Pre-processing

Table 8.1 notably shows the number of missing values for each clinical score. We have approximately the same proportion of missing values both for AD_{TPJ} and AD_t classes (according to the Y_M labelling). Indeed, we count 120 missing values for the AD_{TPJ} class, which corresponds to a ratio of $\frac{120}{43} \simeq 2.79$ missing values per AD_{TPJ} instance, whereas we have 440 missing values for the AD_t class representing a ratio of $\frac{440}{164} \simeq 2.68$ missing values per AD_t sample.

In SCIKIT LEARN, they deal with missing values by replacing them by a value representative of the non-missing data. Three different strategies are possible: the mean, the median or the mode of the known feature values. The optimal strategy depends on the problem and the number of missing values. For some clinical scores, more than half of the values are missing. We simply remove the four features with the highest proportion of missing values (i.e. 125 missing values): DSFw, DSBkw, FluV and Digit. It makes no sense to keep them because more than half the patients would end up with the same synthetic score.

Supervised learning

We assess a forest of 500 Balanced Extra-trees (both with $K = \sqrt{m}$). The Balanced Extra-trees algorithm is the adapted version of Balanced Random Forests [Chen et al., 2004], with Extra-trees instead of Random Forests. This method has been designed for unbalanced datasets. In this method, bootstrap sampling is achieved separately for minority and majority classes in order to randomly sample for each tree a number of instances in the majority class equal to the number of instances sampled in the minority one.

We evaluate the performance of the classifier with a ten repeated leave-one-out cross validation procedure. For each of the ten runs, we compute the importance scores representative of the full dataset. We then average the importance scores over the ten runs to obtain a final vector of importance scores.

If some clinical scores are highly correlated, the risk is to obtain importance scores equally distributed between these scores. They could thus appear less important that another variable important alone. To verify such effect, we analyse correlations between features.

8.4 Results

We first estimate the labelling of the ADNI database. In a second step, we determine the clinical scores that are relevant to our problem.

Table 8.2 – Performances (AUC) of an Extra-trees (ET) classifier with reverse learning and by comparison with visual labels Y_M. AUC are averaged on 50 runs.

Learning dataset Testing dataset Learning algorithm	ADNI with Y_{ET} CRC_2 ET	ADNI with Y_{ET} vs. Y_M	ADNI with Y_M CRC_2 ET
All features	82.57 ± 0.63	77.95 ± 1.16	81.73 ± 1.66
Brain regions	86.11 ± 0.36	82.52 ± 0.48	89.02 ± 0.81

8.4.1 ADNI labelling

The ADNI labelling is composed of two main stages: the group selection procedure and therefore the proper labelling using a classifier learnt from the CRC_2 database.

Group selection

The ensemble classifier with the group selection shows an accuracy of $71.73(\%) \pm 2.87(\%)$. The CER^r group selection method highlights eleven regions of interest that have been selected more than half the time (over the ten repeated ten fold cross validation). These regions are: the rolandic operculum (left), the superior parietal gyrus (left and right), the angular gyrus (right), the precuneus (left and right), the heschl gyrus (left and right), the superior temporal gyrus (left), the middle temporal gyrus (right) and the inferior temporal gyrus (right).

These regions are consistent with results obtained with an SPM analysis comparing the two groups. Indeed, such analysis highlighted an hypo-metabolism in the precuneus region for the typical patients while the TPJ ones were characterized by an hypometabolism in different parietal and temporal areas in comparison with typical individuals.

Classification evaluation

Using the CRC_2 database, we learnt an ET classifier (with or without feature reduction) in order to predict ADNI labels. We therefore evaluate here the consistency of these predicted labels. Table 8.2 provides such estimation with reverse learning but also by comparison with the labels provided by a medical expert.

The reverse learning approach is proposed to estimate the consistency of the labels when no ground truth is available. The aim is to predict the original CRC_2 labels from a classifier learnt using ADNI labels. The second column corresponds to the performance of a classifier learnt on the ADNI database with Y_{ET} labels while the fourth column concerns a classifier based on Y_M labels. The third column simply compares Y_{ET} and Y_M.

The feature selection stage is apparently helpful to increase classifier efficiency. Moreover, a model learnt on visual labels (column 4) provides better AUC values than a model learnt with Y_{ET} labels (column 2) when feature selection is used for both. Column 3 studies relevance between labels obtained with machine learning and with visual inspection. As the AUC values are different from 100%, we can conclude that Y_{ET} do not perfectly match Y_M. Indeed, only 34 samples have been classified as AD_{TPJ} in both approaches. Moreover, the machine learning classifier classified in total 86 AD_{TPJ}

Table 8.3 – Accuracy, sensitivity, specificity and AUC for Balanced Extra-trees classifier ($T = 500$ and $K = \sqrt{m}$) for the three different strategies. ET_m, ET_{med} and ET_{mod} stand for ET with mean, median and mode strategy respectively.

	Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
Y_{ET}	ET_m	56.52 ± 0.94	64.25 ± 1.91	50.92 ± 1.64	60.96 ± 1.07
	ET_{med}	56.43 ± 1.32	63.91 ± 1.73	51.00 ± 1.70	60.99 ± 0.66
	ET_{mod}	57.44 ± 1.43	63.68 ± 2.98	52.92 ± 1.43	61.88 ± 0.80
Y_M	ET_m	55.85 ± 1.53	63.26 ± 3.43	53.90 ± 1.68	63.07 ± 0.71
	ET_{med}	55.85 ± 1.02	61.63 ± 3.51	54.33 ± 1.64	62.73 ± 1.30
	ET_{mod}	56.43 ± 1.57	64.19 ± 2.94	54.39 ± 1.81	63.88 ± 0.67

samples while the medical expert only labelled 43 as AD_{TPJ}. In both cases, labels are highly unbalanced.

8.4.2 Clinical scores detection

In this subsection, we take interest in the clinical scores potentially linked to the existence of two distinct AD metabolic types. We only focus on the ADNI database, using the clinical scores as attributes. Two output vectors are available: the predicted labels Y_{ET} obtained with an ET classifier in combination with a group selection approach, and the medical expert labels Y_M.

Performance

Table 8.3 reports the performance of classifiers for each of the three strategies dealing with missing data. We display accuracy, sensitivity, specificity and area under ROC curve (AUC) values in the table.

Performances are slightly higher than random chance. The classification of AD_t and AD_{TPJ} patients based on their clinical scores is apparently not an easy task. All patients belong to the same broad AD group, for which clinical and neuropsychological assessment are per definition relatively similar. Therefore, the clinical differences between the two classes could be tiny and explain such results.

AUC values are slightly better using Y_M labels than Y_{ET} labels. For both labels, the modal strategy provides accuracy, sensitivity, specificity and AUPR values a little bit higher than the other strategies.

Rank analyses

Table 8.4 provides the ten most important clinical scores identified with each strategy and for each labelling procedure (machine learning vs. medical expert). Importances were averaged over the ten repeats to obtain these rankings.

For Y_{ET}, the clinical scores included in the first six scores are identical for all strategies, i.e. ClockD, TMTA, ADAS_Q1, RAVLTi, ADAS_Q7 and BNT. Until the sixth score, the mean and the median strategies exhibit exactly the same ranking. The only difference for the modal strategy is the inversion of the order for the first two scores. After the second rank, importance scores are decreasing slowly. ADAS_Q8, FAQ and RAVLTl are also common to all strategies.

Table 8.4 – Ranking of clinical scores for Y_{ET} and Y_M. We use short notations to denote each clinical score.

	Y_{ET}					
	ET_m		ET_{med}		ET_{mod}	
	Importance	Clinical score	Importance	Clinical score	Importance	Clinical score
1	$5.04\ 10^{-2}$	ClockD	$5.15\ 10^{-2}$	ClockD	$5.36\ 10^{-2}$	TMTA
2	$4.98\ 10^{-2}$	TMTA	$5.03\ 10^{-2}$	TMTA	$5.06\ 10^{-2}$	ClockD
3	$4.54\ 10^{-2}$	ADAS_Q1	$4.49\ 10^{-2}$	ADAS_Q1	$4.46\ 10^{-2}$	ADAS_Q1
4	$4.26\ 10^{-2}$	RAVLTi	$4.31\ 10^{-2}$	RAVLTi	$4.23\ 10^{-2}$	RAVLTi
5	$4.15\ 10^{-2}$	ADAS_Q7	$4.15\ 10^{-2}$	ADAS_Q7	$4.10\ 10^{-2}$	ADAS_Q7
6	$4.00\ 10^{-2}$	BNT	$4.01\ 10^{-2}$	BNT	$3.95\ 10^{-2}$	BNT
7	$3.88\ 10^{-2}$	ADAS_Q8	$3.89\ 10^{-2}$	FAQ	$3.93\ 10^{-2}$	ADAS_Q8
8	$3.86\ 10^{-2}$	FAQ	$3.88\ 10^{-2}$	ADAS_Q8	$3.83\ 10^{-2}$	RAVLTf
9	$3.79\ 10^{-2}$	RAVLTl	$3.73\ 10^{-2}$	RAVLTf	$3.82\ 10^{-2}$	FAQ
10	$3.75\ 10^{-2}$	RAVLTf	$3.70\ 10^{-2}$	RAVLTl	$3.80\ 10^{-2}$	TMTB
	Y_M					
	ET_m		ET_{med}		ET_{mod}	
	Importance	Clinical score	Importance	Clinical score	Importance	Clinical score
1	$6.57\ 10^{-2}$	TMTA	$6.57\ 10^{-2}$	TMTA	$6.97\ 10^{-2}$	TMTA
2	$4.80\ 10^{-2}$	ADAS_Q3	$4.86\ 10^{-2}$	ADAS_Q3	$4.76\ 10^{-2}$	ADAS_Q3
3	$4.51\ 10^{-2}$	ClockD	$4.57\ 10^{-2}$	ClockD	$4.53\ 10^{-2}$	ClockD
4	$4.16\ 10^{-2}$	ADAS_Q1	$4.20\ 10^{-2}$	BNT	$4.25\ 10^{-2}$	BNT
5	$4.15\ 10^{-2}$	BNT	$3.98\ 10^{-2}$	RAVLTl	$3.97\ 10^{-2}$	ADAS_Q1
6	$3.99\ 10^{-2}$	RAVLTl	$3.90\ 10^{-2}$	ADAS_Q1	$3.93\ 10^{-2}$	RAVLTl
7	$3.75\ 10^{-2}$	CDR	$3.72\ 10^{-2}$	CDR	$3.80\ 10^{-2}$	NPI
8	$3.63\ 10^{-2}$	TMTB	$3.65\ 10^{-2}$	LMIR	$3.69\ 10^{-2}$	CDR
9	$3.59\ 10^{-2}$	LMIR	$3.63\ 10^{-2}$	ADAS_Q8	$3.62\ 10^{-2}$	TMTB
10	$3.55\ 10^{-2}$	ADAS_Q6	$3.60\ 10^{-2}$	TMTB	$3.61\ 10^{-2}$	ADAS_Q8

For Y_M, the first six clinical scores are also similar among all strategies. However, they differ a little bit from those cited above. The scores in the top six are TMTA, ADAS_Q3, ClockD, ADAS_Q1, BNT and RAVLTl. TMTA, ADAS_Q1, ClockD and BNT are thus common to all strategies and both labelling approaches. Using Y_M, we also note that TMTA shows an importance value much higher than the other features.

Figure 8.1 represents box plots for each clinical score normalized between 0 and 1. These box plots allow a comparison of the variables and an analysis of the feature behaviours depending on the class. Scores are ranked according to the mode strategy combined with Y_M. In this figure, the extreme left is the clinical score of lowest rank (highest importance) and the one at the extreme right is the clinical score of highest rank (lowest importance). As we have slightly better performance with the mode strategy with Y_M, we only show this configuration. However, it does not have an influence here as we are just looking at the input values.

Let us focus on the four clinical scores commonly identified by all approaches in the top six, i.e. TMTA, ClockD, BNT and ADAS_Q1. TMTA and ADAS_Q1 show a higher dispersion and a slightly higher median for the AD_t class than for the AD_{TPJ} class. The opposite behaviour is observed for BNT but with a lower effect. ClockD has a higher median value for the AD_{TPJ} class than for the AD_t class.

Random Forests are independent of scaling and normalization. However, they are biased towards discrete variables with more distinct values (or continuous variables) compared to other discrete variables [Strobl et al., 2007]. TMTA is a variable with many distinct values whereas ClockD, ADAS_Q1 and BNT take less values in the sample. The TMTA score could have reached the first place only because of this bias. To verify or refute this assumption, we randomly shuffled the labels (ten times) and looked at the new average ranking over the ten runs. Results are reported in Table 8.5 for the modal configuration with Y_M. The rankings are clearly different. The bias brings TMTA and

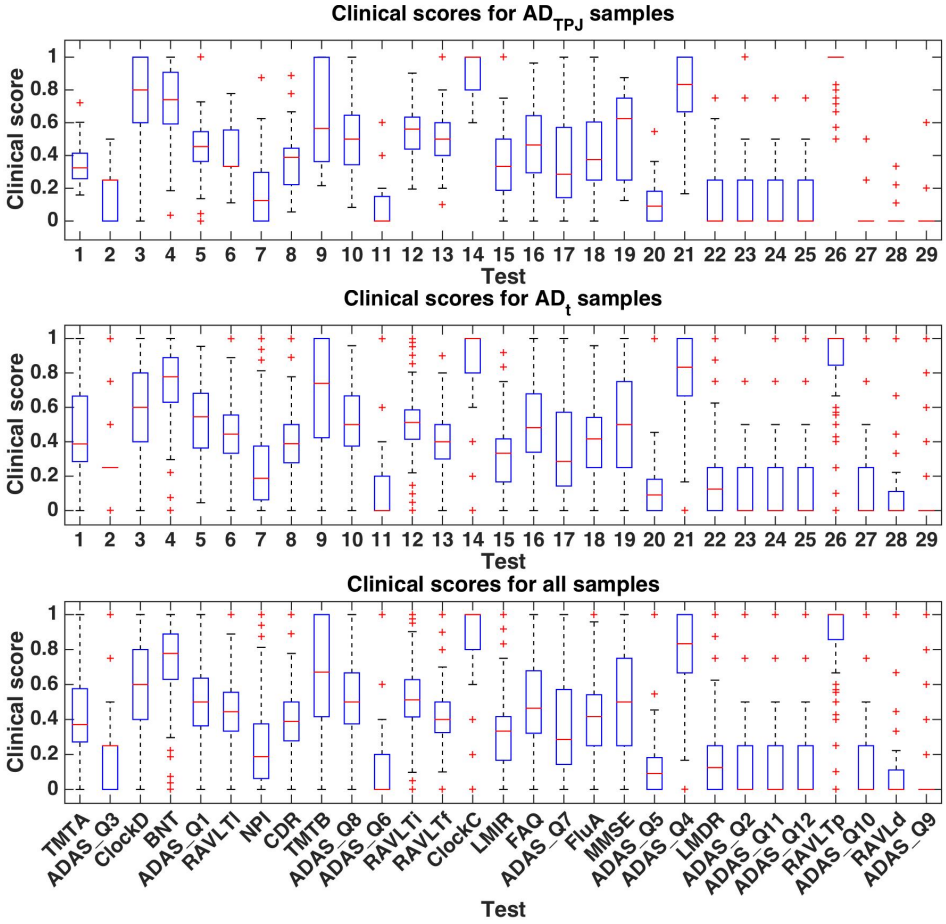


Figure 8.1 – Box plots of the normalized clinical scores in the order they are ranked for the mode strategy with Y_M.

TMTB in the top 5 features but not at the best place.

Finally, the importance of the TMTA test to discriminate the subjects was further confirmed with a t-test, with a performance being significantly ($p < .005$) greater in the TPJ group ($\mu = 57.97$, $\sigma = 29.2$) than the AD$_t$ group ($\mu = 73.23$, $\sigma = 40.17$).

Correlation between variables

Each neuropsychological test evaluates a certain brain network. The different dysfunctions highlighted by clinical scores may sometimes overlap each other. Therefore high correlation could exist between scores and impact the ranking. If two scores were highly correlated, their information would be distributed more or less equally between each other giving rise to similar importance value.

In order to analyse this effect, we thus compute the Pearson correlation coefficient between each pair of clinical scores. Clinical scores with a high proportion of missing values are discarded for the computation, as well as remaining instances with missing values. Figure 8.2 illustrates the correlation matrix.

This figure shows a low positive correlation between most of the ADAS scores. Among the ADAS scores, ADAS_Q10 and ADAS_Q11 are the most highly correlated

Table 8.5 – Ranking of clinical scores for the mode strategy. Comparison between the ranking provided with the original labels and with the randomly shuffled labels. We use short notations to denote each clinical score.

	Original Y_M		Random Y_M	
	Importance	Clinical score	Importance	Clinical score
1	$6.97\ 10^{-2}$	TMTA	$4.18\ 10^{-2}$	FAQ
2	$4.76\ 10^{-2}$	ADAS_Q3	$4.05\ 10^{-2}$	NPI
3	$4.53\ 10^{-2}$	ClockD	$3.97\ 10^{-2}$	ADAS_Q8
4	$4.25\ 10^{-2}$	BNT	$3.94\ 10^{-2}$	TMTB
5	$3.97\ 10^{-2}$	ADAS_Q1	$3.93\ 10^{-2}$	TMTA
6	$3.93\ 10^{-2}$	RAVLTl	$3.91\ 10^{-2}$	CDR
7	$3.80\ 10^{-2}$	NPI	$3.90\ 10^{-2}$	RAVLTi
8	$3.69\ 10^{-2}$	CDR	$3.88\ 10^{-2}$	BNT
9	$3.62\ 10^{-2}$	TMTB	$3.86\ 10^{-2}$	MMSE
10	$3.61\ 10^{-2}$	ADAS_Q8	$3.83\ 10^{-2}$	ADAS_Q7

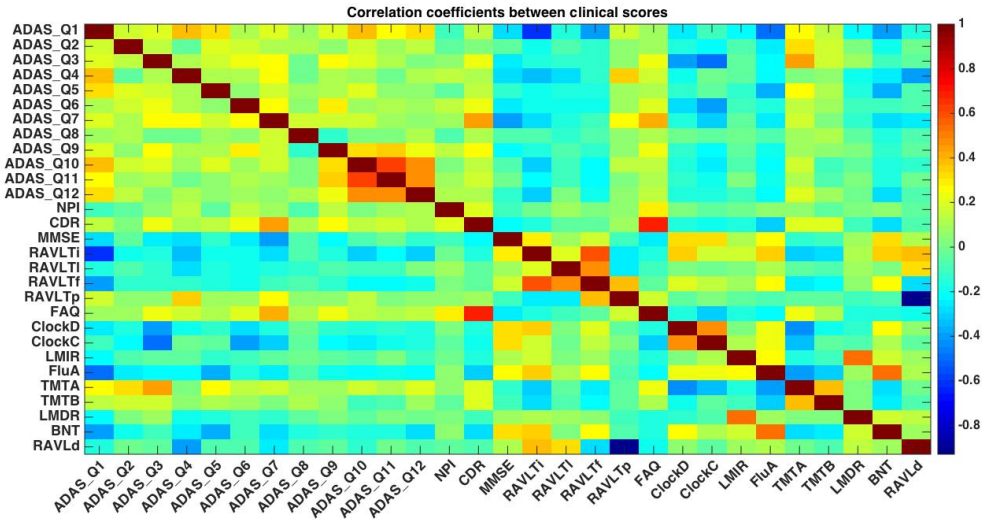


Figure 8.2 – Pearson correlation matrix.

scores. We also observe a positive relatively high correlation between CDR and FAQ scores and between RAVLTi and RAVLTf. The scores RAVLTi and ADAS_Q1 show a relatively high negative correlation. However, the pair of highest negative correlation is composed of RAVLTd and RAVLTp scores. Indeed, with a correlation coefficient of -0.9, it is the only pair with a absolute correlation value higher than 0.75. However, the addition of their importance score to create a new one brings them only to the sixth or seventh position (depending on the strategy, cf. Table 8.4) in the ranking. This correlation is thus not critical to consider.

8.5 Discussion

In this chapter, we tried to characterize two types of Alzheimer's disease patients with neuropsychological data. These types of AD were visually observed in a small PET scan database (CRC_2) in which neuropsychological information was not available. The challenge was to find clinical evidence of different Alzheimer's disease types in the ADNI public database. In this database, all AD patients underwent a high number of clinical and neuropsychological tests.

In this study, the first stage was to learn a tree-based model from the CRC_2 dataset. This model allowed us to predict ADNI labels. The performance of such classifier were assessed by two approaches: a reverse learning procedure and a simple comparison with the labels obtained by a medical expert during the course of the study. Feature selection improves the efficiency of the classifier. The visual labelling appears slightly more accurate than machine learning labelling.

We finally determined clinical scores in the ADNI database by learning a tree-based ensemble model on a learning set composed of clinical scores as features and Y_{ET} vs. Y_M labels as output. We dealt with missing values using three different approaches: the replacement of the missing values by the mean, by the median or by the mode. We obtained a poor performance for all classifiers. The best classifier was the one obtained using Y_M labels in combination with a mode strategy. By analysing relevance scores provided by tree ensembles, we identified neuropsychological characteristics potentially linked to the different types of AD patients, i.e. TMTA, ClockD, BNT and ADAS_Q1. In particular, the TMTA score showed a higher importance value than any other score when Y_M labels were used. Its median and dispersion are quite distinct depending on the class. The TPJ group obtained a performance significantly higher for this test than the typical group. These results suggest that the visuo-motor executive functions are more preserved in the AD_{TPJ} patients. A fMRI study analysing brain activity in healthy subjects performing the TMTA test highlighted mainly brain activity in motor, premotor and visual areas [Karimpoor et al., 2017]. Finally, CRC researchers observed some correlations between metabolism in premotor regions and in the left precuneus for the typical subjects could be responsible for a lower performance of the TMTA test.

Chapter 9

Benchmarking of methods for Alzheimer's disease

Chapter overview

In this chapter, we consider different classification problems related to Alzheimer's disease progression. The main goal is to compare different machine learning methods in terms of performance and interpretability. While we mainly focus on tree-based machine learning approaches in the whole manuscript, we consider here alternative machine learning methods for pattern recognition for Alzheimer's disease. We evaluate linear methods such as SVM and Lasso, which are frequently used in the neuroimaging field, but also non linear methods such as Random Forests. We evaluate each method in their feature-based version vs. their group-based version.

9.1 Problem definition

Alzheimer's disease shows different stages of evolution: the cognitively normal stage, the mild cognitive impairment stage and finally dementia. The evolution of the disease through these different stages is characterized by changes in biomarkers. Clinically speaking, neuropsychological assessments will give different results depending on the time line. Moreover, disease evolution will cause abnormal brain atrophy and hypometabolism in specific brain regions. These consequences of the disease will notably be observable with MRI images and PET images respectively.

We consider here three distinct problems: the diagnosis of demented patients (from very mild to mild) against normal individuals from MRI data, the prognosis of MCI patients from PET images, and the differentiation of MCI and AD patients from PET data. Data used to deal with these problems are respectively the OASIS dataset, the CRC dataset, and the $ADNI_2$ dataset. They were presented in Subsection 3.3.

Structural differences between mild demented patients and normal individuals have already been studied a lot in the literature through classification frameworks. This task appears to be relatively easy to undertake as it provides in general good classification performance with simple frameworks. For instance, Klöppel et al. [2008] proposed to use a linear SVM-based classifier to distinguish AD and cognitively normal individuals from gray matter images. They showed high performance and robustness of the method for this task. More precisely, they achieved nearly 90% of accuracy. Moreover, they per-

formed analyses using scans from different centers for training and testing respectively and showed good performance of their classifier. They obtained best results with whole brain features. They also argued that non linear kernel did not provide better results. In [Magnin et al., 2009], the authors proposed to deal with this problem using SVM and extraction of regions of interest (ROI) from gray matter images. With a small database of only 38 samples, they showed a classification accuracy of 94.5%. Schrouff et al. [2018] studied the same classification task with the OASIS database and an MKL approach. In this approach, one kernel is built from each region of interest defined by an atlas. Such method attributes a weight to each region as a SVM attributes a weight to each feature. They achieved an accuracy lower than 70% and did not beat the performance of SVM except for one atlas.

The prognosis of MCI from neuroimages is a question that has also been investigated deeply in the literature, e.g. in [Zhang et al., 2012, Moradi et al., 2015, Gray et al., 2013] to cite just a few. Such problem is less obvious to solve than the previous one and classifiers show therefore in general lower performance. In [Zhang et al., 2012], they combined multiple modalities (MRI, FDG-PET and cerobrospinal fluid data) in order to obtain the best accuracy possible with a classifier called *multi-modal SVM*, which consists basically in a MKL approach with a kernel for each modality. At the end, they reached about 74% of accuracy for the classification of MCI converters vs. non converters. Moradi et al. [2015] used MRI data and feature selection (based on a regularized logistic regression) to build a Random Forests classifier. They provided a detailed study by comparing their approach with other state of the art methods but also by giving a list of publications made in the field for the AD conversion based on the ADNI database. Depending on the samples, the validation method and the machine learning algorithm, results can vary a lot from around 60% accuracy to more than 80%. Finally, in [Gray et al., 2013], the authors also worked with the ADNI database, using a combination of multiple biomarkers (MRI, PET, CSF and genetics), in order to distinguish converters and non converters. They used a Random Forests algorithm for classification but also to obtain similarity measures between instances for each biomarker. They reached no more than 58% accuracy with their multi-modality similarity approach for the classification of MCIs vs. MCIc while they obtained satisfying results for the classification of AD vs. CN and MCI vs. CN.

Metabolism differences between MCI and AD patients can be observed through PET data. The evolution of brain energy consumption across the disease stages is also a question that matters in research. MCI vs. AD classification task has notably been studied with the ADNI database in [Segovia et al., 2015]. They used a MKL method in combination with feature extraction approach such as Principal Component Analysis, Non-Negative Matrix Factorization and Haralick procedure. Their classifier achieved nearly 80% of accuracy.

In conclusion, many methods have already been used to answer these research questions. However, results are generally difficult to compare from one paper to another as images often came from different origins, different pre-processing parameters were chosen, or different assessment procedures were used. In this chapter, we thus propose a benchmarking of several machine learning methods.

9.2 Supervised learning

We compare several supervised learning methods from different perspectives: linear vs. non linear approach and feature-based vs. group-based approach. Each method will be investigated from an accuracy and an interpretability point of view.

Table 9.1 – Characteristics of the 6 methods considered in this chapter.

Method \ Characteristic	Linear	Type	Sparse	Group-based
SVM	✓	Kernel		
MKL	✓	Kernel	✓	✓
Lasso	✓	Regularization	✓	
Group Lasso	✓	Regularization	✓	✓
Random Forests		Tree	✓	
Group Random Forests		Tree	✓	✓

9.2.1 Methods

Table 9.1 presents the different methods used in this chapter and provides their main characteristics. Each type of method will be studied in both versions: feature-based vs. group-based approach. Except for the last one, all methods have been introduced in Chapter 2. The last method, called *Group Random Forests*, is an adapted version of Random Forests which takes into account data structure. It has already been explained in Section 5.5. Instead of looking for the best features among K from the whole feature set at each node, K groups are randomly drawn with replacement and one feature in each group is subsequently randomly selected. The best feature is then chosen among this new set of K features. This way, features from a small group are as likely to be selected as feature from a large group. For neuroimaging data, it thus considers that a small brain region is as important to analyse as a larger one.

We use MATLAB packages to test all methods. In particular, SVM and MKL methods are available in the PRONTO toolbox[1] [Schrouff et al., 2013b]. In the toolbox, only linear kernels are provided. Random Forests algorithm is provided by the RT package[2]. Finally, Lasso and Group Lasso approaches are estimated through their implementation in the SLEP package[3] [Liu et al., 2009].

9.2.2 Parameter tuning and performance assessment

Methods are assessed by performing ten repeated ten-fold cross validations and averaging the results over the ten runs. Accuracy, sensitivity, specificity and AUC values are computed for each method. Moreover, methods are also compared by plotting their ROC curves.

For linear kernel methods, the C hyper-parameter is optimized inside a nested ten fold cross validation loop for each fold over all the runs (the different values assessed are $C = 10^{[-3:1:3]}$). Features are mean centred and normalized by their standard deviations before training, as proposed in [Schrouff et al., 2018].

For Lasso methods, the regularization parameter λ is also optimized in a nested ten fold cross validation loop for each fold over all the runs. The different values of λ that are assessed are $\lambda = [0.01\ 0.025\ 0.05\ 0.1\ 0.5\ 0.75\ 1] \times \lambda_{max}$ where λ_{max} is, according to [Liu et al., 2009], the maximal value of λ above which the objective function of the optimization problem (see Equations (2.34) and (2.36) in Chapter 2 for the reminder) will be null (i.e. all weights equal to zero). The value of λ_{max} is automatically computed in the SLEP

[1] http://www.mlnl.cs.ucl.ac.uk/pronto/
[2] http://www.montefiore.ulg.ac.be/~geurts/Software.html
[3] http://www.yelab.net/software/SLEP/

package.

For tree-based methods, the parameter K is also optimized through a nested ten fold cross validation loop. We propose to assess the values $K = [\sqrt{m}\ 1000\ 10,000]$ to observe the effect of increasing K from the default value. The value of K for the group-based approach of Random Forests consists in randomly picking with replacement K groups and then randomly one feature in each group. Forests of $T = 1000$ trees are fitted to compare feature-based vs. group-based approach.

For methods involving group structure, we consider feature divisions defined by the AAL atlas (116 brain areas).

9.2.3 Model interpretation

SVM and Lasso methods provide feature weights while MKL directly attributes weights per region. Indeed, in MKL, each brain region (delimited by an atlas) has its features directly associated to a kernel. The selection is then achieved by considering each group of features as an entity. With such method, the interpretability is easy as weight maps per region can be directly generated. As already shown in previous chapters, we can also analyse a ranking of regions by sorting brain areas from the highest weight to the lowest.

In order to obtain rankings of brain areas for SVM and Lasso methods, we thus aggregate feature weights inside brain regions defined by the same atlas such that

$$W(R) = \frac{\sum_{i \in R} |w_i|}{N(R)}, \tag{9.1}$$

where $W(R)$ is the weight of region R composed of $N(R)$ features and w_i is the weight of feature i attributed by the method, as in [Schrouff et al., 2013a].

Importance scores provided by tree-based methods are averaged in a similar way, as already proposed in previous chapters.

All scores are averaged over the ten repeated ten fold cross validation procedure and then aggregated using the AAL atlas.

9.3 Results

9.3.1 OASIS dataset

Performance

Accuracy, sensitivity, specificity and AUC values are displayed in Table 9.2. Linear methods outperform tree-based methods for nearly all performance measures. MKL is the least efficient linear method and shows a similar AUC value as RF approaches but a better accuracy, sensitivity and specificity. All methods show a better specificity than sensitivity. This is what we are looking for, as in such problem it is important not to miss an AD diagnosis. Figure 9.1 shows ROC curves of all methods. SVM methods show a high true positive rate at the expense of a high false positive rate. For a relatively low false positive rate, Lasso methods provide the best true positive rate and curves are quite similar both for feature-based and group-based approaches. Finally, tree-based methods are the best in the only case for which nearly no false positive value is tolerated. Both tree based ensemble curves are almost coinciding.

Table 9.2 – OASIS dataset. Accuracy, sensitivity, specificity and AUC for all methods.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
SVM	68.10 ± 1.85	67.00 ± 2.36	69.20 ± 3.68	74.50 ± 1.53
MKL	69.00 ± 2.91	65.40 ± 3.13	72.60 ± 4.72	71.05 ± 1.98
Lasso	69.70 ± 2.87	67.20 ± 3.43	72.20 ± 4.66	73.16 ± 2.24
Gp Lasso	70.20 ± 1.87	68.40 ± 1.58	72.00 ± 3.27	73.48 ± 2.04
RF	64.00 ± 1.33	57.40 ± 2.32	70.60 ± 2.12	70.22 ± 1.26
Gp RF	65.30 ± 1.49	59.80 ± 2.57	70.80 ± 1.40	71.41 ± 1.06

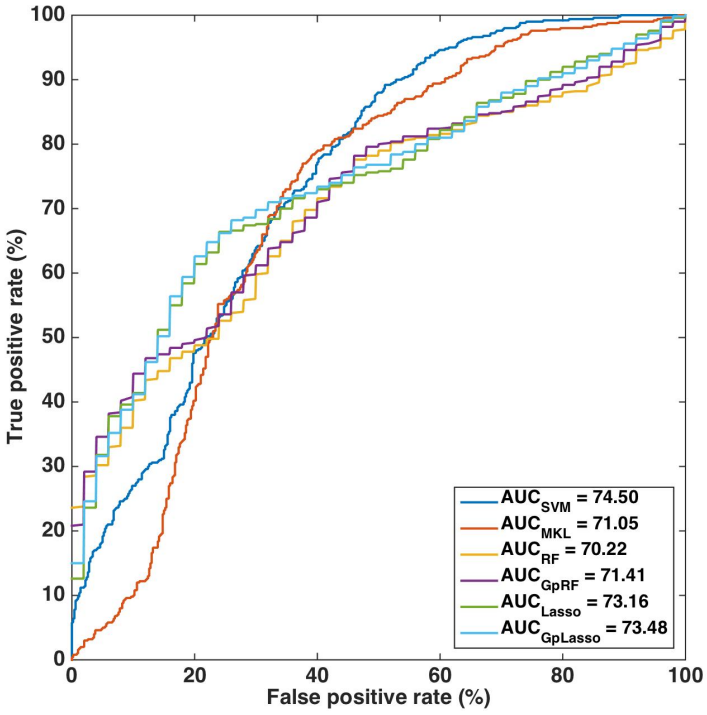


Figure 9.1 – OASIS dataset. Roc curves of all methods.

Figure 9.2 represents the selection frequency of optimized parameters for each method. For kernel methods, the parameter value of highest frequency is 0.1 for SVM and 100 for MKL. The value frequencies for Lasso and Gp Lasso are quite distributed among all possible values of λ, with the highest frequency for $\lambda = 0.01$ for both methods and then decreasing frequency with increasing parameter value. We observe a similar behaviour for tree-based methods. The parameter value the most chosen is $\sqrt{m}$ and the frequency of selection decreases with an increase of K.

Interpretability

The ten top-ranked regions with each method are provided in Table 9.3. Depending on the chosen approach (feature-based vs. group based) ranking can change a lot. We observe that RF method is the most consistent from this point of view, as RF and Gp RF

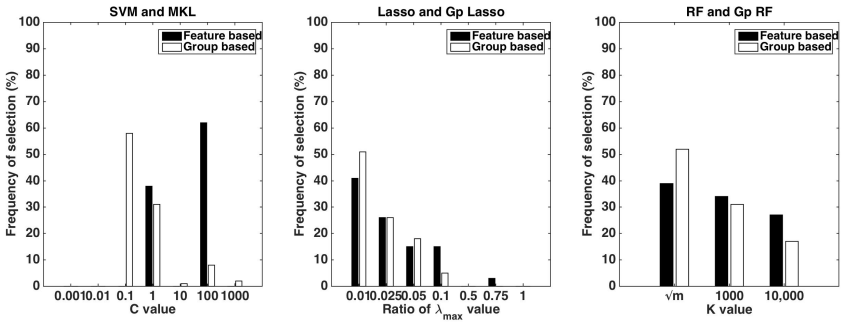


Figure 9.2 – OASIS dataset. Parameter optimization.

Table 9.3 – OASIS dataset. The ten most contributing AAL regions for each method. L, resp. R, stands for left, resp. right, hemisphere. Regions common to at least three methods are highlighted in bold.

Feature-based			
Rank	SVM	Lasso	RF
1	**Cerebelum 10 (R)**	**Hippocampus (R)**	**Hippocampus (R)**
2	Cerebelum Crus2 (R)	**Hippocampus (L)**	Amygdala (R)
3	Cerebelum Crus2 (L)	Temporal Pole Sup (R)	Amygdala (L)
4	Paracentral Lobule (L)	Putamen (L)	**Hippocampus (L)**
5	**Frontal Sup (L)**	**Thalamus (L)**	ParaHippocampal (R)
6	Precuneus (L)	**Frontal Sup (L)**	**Thalamus (L)**
7	Precuneus (R)	Amygdala (R)	Temporal Mid (L)
8	Cerebelum 8 (R)	**Cerebelum 10 (R)**	Occipital Inf (L)
9	Supp Motor Area (R)	ParaHippocampal (R)	ParaHippocampal (L)
10	Cerebelum 7b (L)	Cerebelum Crus1 (L)	Temporal Mid (R)
Group-based			
Rank	MKL	Gp Lasso	Gp RF
1	**Hippocampus (R)**	Paracentral Lobule (L)	Cerebelum 10 (L)
2	**Thalamus (L)**	Paracentral Lobule (R)	**Cerebelum 10 (R)**
3	Frontal Inf Tri (L)	Parietal Sup (R)	Amygdala (R)
4	Lingual (L)	Supp Motor Area (R)	Amygdala (L)
5	**Frontal Sup (L)**	Supp Motor Area (L)	**Hippocampus (R)**
6	Temporal Inf (L)	**Cerebelum 10 (R)**	Vermis 10
7	Frontal Inf Oper (R)	Parietal Sup (L)	**Hippocampus (L)**
8	**Hippocampus (L)**	Frontal Sup (R)	Pallidum (L)
9	Cerebelum Crus1 (L)	Precuneus (L)	Pallidum (R)
10	Precuneus (L)	Postcentral (R)	Cerebelum 3 (L)

have four common regions in the top-ten while Lasso methods have only one common region between feature-based and group-based approaches and kernel methods only two. Although Group Lasso provided slightly better results in terms of performance, Lasso identifies more accurately the regions relative to the phenotype of interest. Indeed, as already stated in Chapter 6, we expect to find with ML methods brain areas mainly related to the hippocampus [Gosche et al., 2002, Klöppel et al., 2008].

9.3.2 CRC dataset

Performance

Accuracy, sensitivity, specificity and AUC values are displayed in Table 9.4. In terms of accuracy, specificity and AUC values, Lasso and tree-based approaches outperform

Table 9.4 – CRC dataset. Accuracy, sensitivity, specificity and AUC for all methods.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
SVM	67.33 ± 5.55	67.27 ± 4.69	67.39 ± 10.70	73.58 ± 5.85
MKL	61.11 ± 7.43	59.09 ± 7.42	63.04 ± 13.16	64.33 ± 8.42
Lasso	69.33 ± 3.44	67.73 ± 5.00	70.87 ± 6.17	75.14 ± 3.83
Gp Lasso	70.00 ± 3.81	68.64 ± 4.52	71.30 ± 3.67	79.29 ± 2.81
RF	70.44 ± 2.11	64.09 ± 4.52	76.52 ± 3.04	77.82 ± 2.72
Gp RF	71.11 ± 4.44	66.82 ± 6.45	75.22 ± 5.44	76.17 ± 3.75

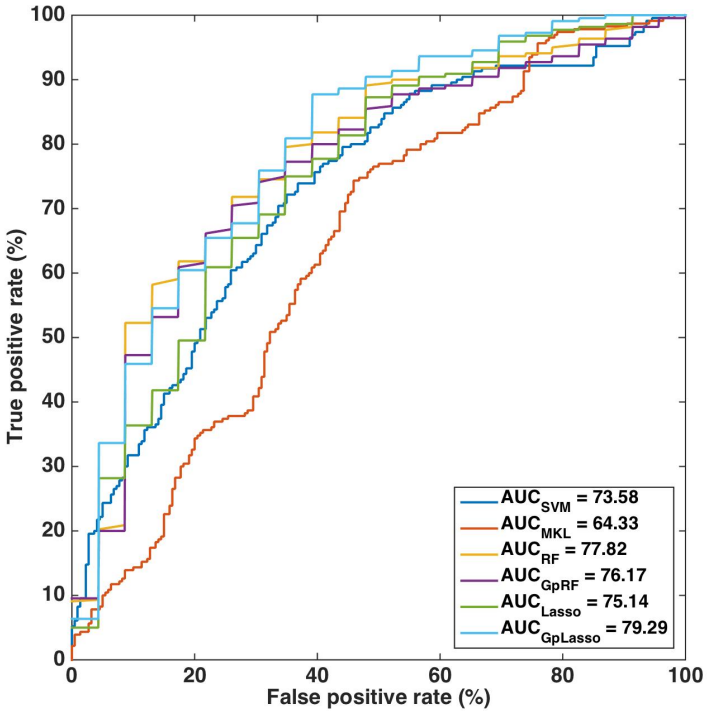


Figure 9.3 – CRC dataset. Roc curves of all methods.

kernel methods. Tree-based methods show the highest specificity values at the expense of a lower sensitivity than most of the other methods. In other words, tree-based methods are better to diagnose non converters with confidence than the other methods and so a high number of people diagnosed as non converters are truly non converting ADs. In our type of problems, we look for high specificity methods in order not to miss an individual who will develop the disease and will thus not be treated because of a wrong diagnosis. MKL is clearly the worst method from all points of view. We also observe this behaviour in Figure 9.3 for ROC curves. This figure mainly illustrates the lower efficiency of kernel methods. Moreover, RF with features is among the best curves for any false positive rate while Group Lasso outperforms it for a high false positive rate.

Figure 9.4 shows the selection frequency of optimized parameters for each method. For SVM, $C = 1$ is the most frequently selected value while it is $C = 100$ for MKL. For

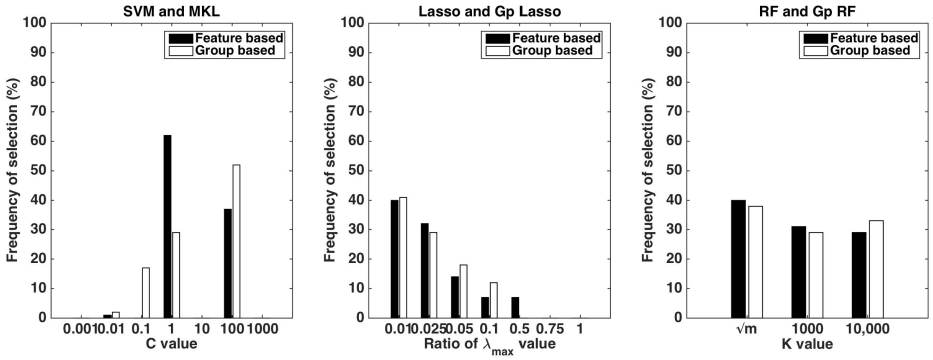


Figure 9.4 – CRC dataset. Parameter optimization.

Lasso methods, the selection frequency decreases with an increase of the parameter value. For both methods, $\lambda = 0.01$ is the value that is the most often selected. For tree-based methods, $K = \sqrt{m}$ is the most often selected value. The other values are selected with a similar frequency.

Interpretability

The ten top-ranked regions with each method are provided in Table 9.5. There is some consistency between top-ranked regions obtained with feature-based approach vs. group-based approach. Indeed, the medial temporal gyrus (right hemisphere), the angular gyrus (right hemisphere) and the inferior parietal gyrus (right hemisphere) are common to most methods. This is consistent with expected results. Indeed, temporoparietal areas are in general observed as a distinctive characteristic for MCI stable versus converters. Although MKL provides significantly lower performance than SVM, both highlight regions expected to explain the prognosis of AD. Additionally, we observe that some methods detect more parietal regions and others more temporal regions. Therefore it appears that methods are complementary and it is hard to vote in favour of one or another.

9.3.3 ADNI$_2$ dataset

Performance

Table 9.6 shows higher accuracies, sensitivities, specificities and AUC values for all methods. Group structure consideration does not have a high impact on performances for Lasso and tree methods. However, for kernel methods, the use of a group approach decreases the method performance. Finally, Figure 9.5 shows that tree-based ensemble methods outperform other methods. Except for MKL, the ROC curves of linear methods appear rather similar.

Figure 9.6 shows the selection frequency of optimized parameters for each method. For SVM, the parameter value the most often selected is $C = 1$ whereas it is $C = 100$ for MKL. We observe higher frequency of selection for $\lambda = 0.1$ and 0.5 with Lasso. For Group Lasso, frequencies are rather distributed among all values between 0.01 and 0.5. For tree-based methods, we see a higher preference for the values $\sqrt{m}$ and 1000 than for $10,000$.

Table 9.5 – CRC dataset. The ten most contributing AAL regions for each method. L, resp. R, stands for left, resp. right, hemisphere. Regions common to at least three methods are highlighted in bold.

Feature-based			
Rank	SVM	Lasso	RF
1	**Parietal Inf (R)**	**Parietal Inf (R)**	**Angular (R)**
2	**Angular (R)**	**Thalamus (L)**	**Temporal Mid (R)**
3	Cerebelum 7b (R)	**Angular (R)**	**Parietal Inf (R)**
4	**Temporal Mid (R)**	Cerebelum 7b (R)	Temporal Mid (L)
5	Paracentral Lobule (L)	**Temporal Mid (R)**	Vermis 7
6	**Vermis 8**	Vermis 10	Cuneus (L)
7	Temporal Inf (R)	Cerebelum Crus2 (L)	**Vermis 8**
8	Temporal Pole Mid (L)	Cerebelum Crus2 (R)	Temporal Inf (R)
9	Cerebelum Crus2 (L)	Thalamus (R)	Cerebelum 8 (L)
10	**Thalamus (L)**	Vermis 7	Temporal Inf (L)
Group-based			
Rank	MKL	Gp Lasso	Gp RF
1	**Temporal Mid (R)**	**Parietal Inf (R)**	Vermis 1 2
2	**Angular (R)**	Paracentral Lobule (L)	**Angular (R)**
3	Vermis 6	Supp Motor Area (R)	Vermis 7
4	**Thalamus (L)**	Parietal Inf (L)	**Parietal Inf (R)**
5	Frontal Sup Medial (R)	Supp Motor Area (L)	**Vermis 8**
6	Cerebelum 10 (L)	Paracentral Lobule (R)	**Temporal Mid (R)**
7	Temporal Mid (L)	Parietal Sup (L)	Cerebelum 3 (L)
8	**Vermis 8**	Precentral (R)	Cerebelum 10 (L)
9	Parietal Sup (R)	Cingulum Mid (L)	Vermis 9
10	Hippocampus (R)	Parietal Sup (R)	Vermis 6

Table 9.6 – $ADNI_2$ dataset. Accuracy, sensitivity, specificity and AUC for all methods.

Method	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
SVM	62.15 ± 1.58	53.62 ± 2.08	69.72 ± 3.13	68.00 ± 1.55
MKL	59.55 ± 1.36	53.51 ± 3.40	64.91 ± 2.03	64.37 ± 1.75
Lasso	61.85 ± 1.45	51.38 ± 2.51	71.13 ± 2.00	66.35 ± 1.23
Gp Lasso	62.35 ± 2.61	54.47 ± 4.19	69.34 ± 2.45	66.70 ± 2.35
RF	66.55 ± 1.54	59.36 ± 2.17	72.92 ± 1.99	73.46 ± 1.22
Gp RF	66.85 ± 1.73	59.68 ± 2.94	73.21 ± 1.79	73.68 ± 0.69

Interpretability

The ten top-ranked regions with each method are provided in Table 9.7. Regions in the top ranks for Random Forests, which was one of the best performers above, are consistent with literature. Indeed, regions the most often detected as relevant to distinguish AD and MCI patients with FDG-PET are notably the posterior cingulate cortex, the hippocampus, the temporal and the pre-frontal areas [De Santi et al., 2001, Drzezga et al., 2003, Mosconi et al., 2008]. With the group approach, we only find three common regions to the ranking provided by Random Forests. Lasso and MKL methods also found some consistent regions of interest. On the contrary, SVM and Gp Lasso highlight a lot of irrelevant regions. The posterior cingulate cortex (right) is detected by all methods except these two. This is also surprising, considering that SVM outperforms MKL in terms of performance. Finally, there is no common region between Lasso and Group Lasso while there are three regions in common between SVM and MKL.

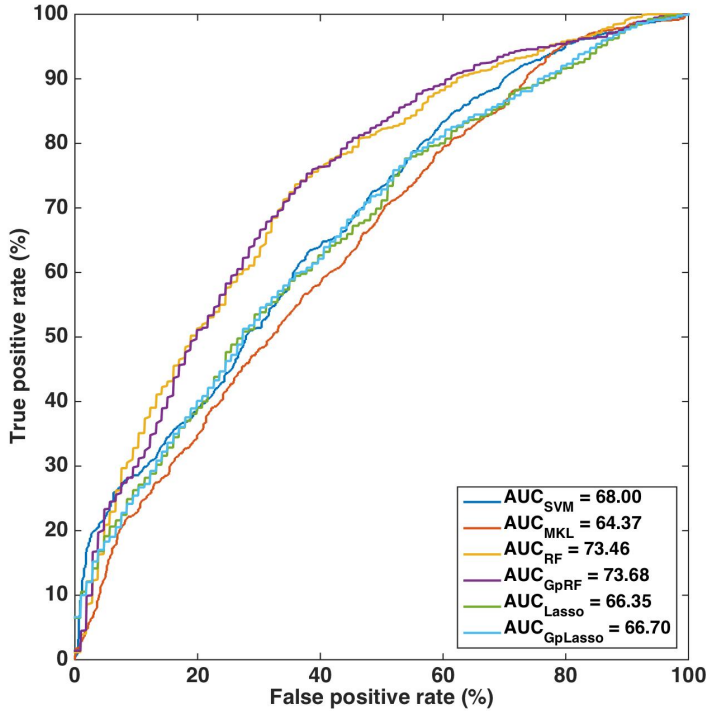


Figure 9.5 – $ADNI_2$ dataset. Roc curves of all methods.

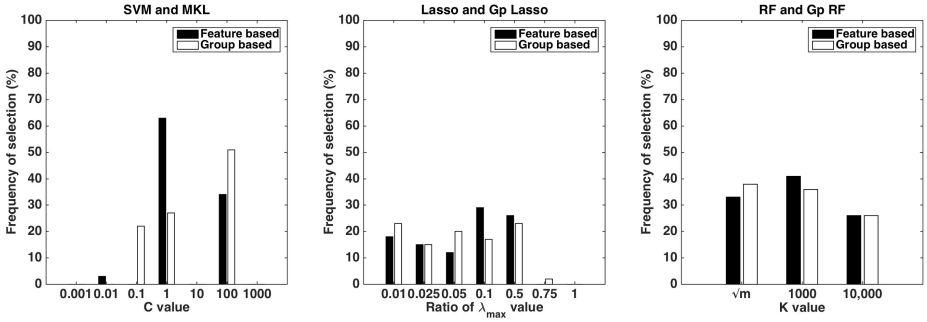


Figure 9.6 – $ADNI_2$ dataset. Parameter optimization.

9.4 Discussion

In this chapter, we studied different research questions about Alzheimer's disease with different machine learning methods and approaches. We identified on each dataset the best method from an accuracy and an interpretability points of view.

Especially, with the OASIS dataset, we identified that linear methods such as SVM, Lasso and Gp Lasso are efficient to distinguish control from mild AD patients. However, while Gp Lasso and SVM showed accurate predictions, their interpretation of the disease was not entirely consistent by comparison with expected results from past literature. Lasso and MKL are more consistent. Random Forests method, which provided a lower accuracy, highlighted more regions relevant to the disease phenotype than SVM and Gp Lasso. All classifiers provided lower performance than most of state of the art

Table 9.7 – $ADNI_2$ dataset. The ten most contributing AAL regions for each method. L, resp. R, stands for left, resp. right, hemisphere. Regions common to at least three methods are highlighted in bold.

Feature-based			
Rank	SVM	Lasso	RF
1	Vermis 8	**Amygdala (R)**	**Angular (R)**
2	Vermis 9	**Cingulum Post (R)**	**Cingulum Post (R)**
3	Cerebelum Crus2 (L)	**Cingulum Post (L)**	Temporal Inf (R)
4	**Amygdala (R)**	**Angular (R)**	**Cingulum Post (R)**
5	Cerebelum Crus1 (L)	ParaHippocampal (L)	Angular (L)
6	Paracentral Lobule (L)	Temporal Pole Mid (L)	Cerebelum Crus2 (L)
7	Supp Motor Area (L)	Vermis 3	Putamen (R)
8	ParaHippocampal (L)	Cuneus (R)	Temporal Mid (R)
9	Temporal Pole Mid (L)	Temporal Sup (L)	Temporal Inf (L)
10	Vermis 7	Cerebelum Crus2 (L)	Parietal Inf (R)
Group-based			
Rank	MKL	Gp Lasso	Gp RF
1	**Angular (R)**	Paracentral Lobule (L)	**Cingulum Post (R)**
2	Temporal Inf (R)	Paracentral Lobule (R)	Vermis 1 2
3	Frontal Sup (R)	Supp Motor Area (L)	Cerebelum 3 (R)
4	Angular (L)	Supp Motor Area (R)	**Cingulum Post (L)**
5	Postcentral (L)	Parietal Inf (L)	Vermis 10
6	**Amygdala (R)**	Parietal Sup (L)	Vermis 7
7	Cerebelum Crus2 (L)	Parietal Inf (R)	Vermis 3
8	Cerebelum Crus1 (L)	Postcentral (R)	**Angular (R)**
9	Precentral (L)	Parietal Sup (R)	Cerebelum 10 (R)
10	**Cingulum Post (R)**	Precentral (R)	Vermis 9

results. However, the OASIS images used for our experiments are composed of AD patients from very mild to mild AD only, which makes the problem harder to solve.

For the CRC dataset, the traditional Random Forests method achieved good accuracy, specificity and AUC values. The best AUC value is however achieved with Gp Lasso. All methods identified several regions related to the evolution of the disease. Feature-based approach and group-based approach allow us to detect different regions of interest. Fortunately, some regions are common to most methods like the angular gyrus (right) and the middle temporal gyrus (right). In comparison with state of the art results, the performance of the classifiers are quite good.

Finally, with the $ADNI_2$ dataset, we analysed functional differences depending on the stage of the disease. To distinguish MCI and AD stages, the best performer was the RF approach. Except for kernel methods, group and feature approaches provided similar results in terms of accuracy. Nevertheless, it is more difficult to conclude regarding the interpretability. Indeed, RF and Gp RF both highlighted relevant regions about brain metabolism evolution while MKL appeared more consistent for kernel methods. On the contrary, for Lasso methods, the group-based approach gave less consistent result in the top-ranked regions than the feature-based one. The classification of MCI and AD patients from their PET scan provided poor performance in comparison with [Segovia et al., 2015] using also PET from ADNI (140 images). In this case, feature extraction approaches helped apparently to achieve better accuracy.

To conclude, we cannot claim in favour of one particular approach compared to another for any problem. However, Random Forests provided good accuracy, specificity and AUC values and also consistent interpretability on any problem. It therefore seems

an approach of choice, while there is no guarantee to obtain the most accurate diagnosis system with this method. In addition, we should advise to test both approaches, i.e. feature-based vs. group-based, as the group-based approach can allow us to detect other patterns of interest without any loss of accuracy.

Part IV

Conclusion and prospects

Chapter 10

Conclusions and perspectives

Question everything. Learn something. Answer nothing.

- Euripides, *c. 480 – c. 406 BC*

The objective of this thesis was first to explore the possibilities that tree-based ensemble methods offer in the field of neuroimaging and second to exploit these methods to improve our understanding of Alzheimer's disease. Our main methodological contributions include an analysis of variable importance scores derived from tree ensembles in the context of high dimensional data and various improvements of these scores to cope with this high-dimensionality by building on specificities of neuroimaging datasets. Our contributions concerning Alzheimer's disease consist in the application of the developed methods on several datasets related to various research questions around this disease.

We describe below our main general contributions and findings throughout the thesis and then discuss some future work directions. We refer the reader to the conclusions of the separate chapters for more detailed discussions.

10.1 Main findings and conclusions

As typical neuroimaging datasets are characterized by a very low number of samples and a very high dimensionality, we started the thesis in Chapter 4 by questioning the validity of variable importance scores in such an extreme setting. We first showed mathematically that the expected number of trees required to have seen all features at least once can be very large for the typical learning sample size and number of features of neuroimaging datasets. Through an empirical study, we furthermore highlighted that many more trees than this minimal number are actually required for importance scores to reach stability. Overall, these observations call for using very large forests when one wants to rely on these importance scores to identify the most relevant features. Our experiments however also show that stability is obviously not enough to ensure that the most important features found through these scores are indeed truly relevant. The problem of identifying a few relevant features among many irrelevant ones remains very challenging especially when only a limited number of samples is available. In such a setting, building even the largest possible forest does not prevent variable importance scores to fail identifying all or even any relevant features.

These somewhat negative results motivated us to explore in Chapter 5 several adaptations of variable importance scores that try to improve the properties of these scores

in the particular case of neuroimaging data. To address issues raised by the high dimensionality, these adaptations all exploit either the 3D spatial organization of the features or a division of these features into groups through a pre-defined atlas. Our experiments on artificial and real datasets clearly show that exploiting such structures allows to improve importance scores, in particular in small sample size settings and in the case of small forest sizes.

One of the most interesting methods highlighted in Chapter 5 turned out to be group based aggregation, which is studied in details in Chapters 6 and 7. The idea of this method is to associate a score to groups of features by aggregating the individual importances of the features in this group. This method is motivated first by the fact that estimating reliably the importance of a group of features is expected to be easier than estimating reliably the importance of an individual feature. Working at the level of groups indeed naturally reduces the dimensionality of the problem. This is confirmed by several experiments in these chapters. Second, reporting group importances instead of feature importances also makes the results more interpretable for medical experts that are more interested in highlighting brain regions than individual isolated voxels.

One limitation of these group importance scores however, which is inherited from feature importance scores, is that they are not really interpretable, which makes difficult finding a threshold above which all groups can be safely considered as truly relevant. To tackle this problem, we adapted at the group level several methods proposed in the literature to turn variable importance scores into more interpretable statistical scores that can then be more easily thresholded. The application of these techniques at the group level has at least two advantages with respect to their use at the feature level. First, statistical power is improved when working at the group level. Second, computing times are also very much reduced. Indeed, these methods require to perform thousands of random permutations for each importance score, whose number is very much reduced when going from features to groups.

Empirically, we found that most of these methods work well in the sense that they indeed allow to identify the truly relevant groups at the top of group importance rankings. On real neuroimaging datasets, only very few groups are identified as truly relevant using these methods but this is not surprising given the challenge that these datasets represent for feature selection techniques (as discussed earlier and in Chapter 4). Interestingly, we also showed that using these techniques to select a few groups prior to growing a Random Forests model can result in an improvement of the predictive performance of this model.

The practical relevance of these techniques was highlighted in these chapters notably on the CRC dataset, where the goal is to distinguish converter from stable MCI patients for the prognosis of Alzheimer's disease. Several brain regions were highlighted by the developed methods, most of them being confirmed in the literature as being involved in the prognosis of Alzheimer's disease. We furthermore showed that it is possible to predict with an error rate close to 20% (as estimated from a balanced dataset by cross-validation). This result is satisfactory, given the very small size of the available dataset and the difficulty of making such prognosis.

In the context of a collaboration with medical doctors, we tackled in Chapter 8 the problem of characterizing Alzheimer's disease subtypes with neuropsychological data. These subtypes of AD were visually observed in a small PET scan database (CRC_2), in which neuropsychological information was not available. The proposed approach was therefore to train a machine learning model on this database in order to predict the subtypes of AD patients in a larger public database in which extensive clinical infor-

mation was provided (ADNI). Then, we used clinical information and inferred labels on the public database to construct a new tree-based model. Through the resulting importance scores, we were able to identify relevant neuropsychological factors, whose biological relevance has been analysed by medical doctors.

Finally, in Chapter 9, we compared, on three different datasets related to Alzheimer's disease, linear vs. non-linear methods and feature-based vs. group-based methods. This empirical analysis showed that Random Forests are competitive with linear methods in terms of predictive performance and that they provide good interpretability in general. Group-based variants of the methods only marginally affect predictive performance however, positively or negatively. In terms of interpretability, there can be important differences in the regions highlighted by different methods, even between two linear methods and between the feature-based and group-based variants of the same method. This is a consequence of the different biases that these methods introduce and also of the very small size of the datasets in comparison with the number of features. This result suggests that it might be useful to consider to apply several methods on any new dataset, either to have more confidence about regions that are confirmed by all methods or, in more exploratory studies, to be sure to not miss any potentially relevant region. In all cases, of course, the medical relevance of the highlighted regions should be validated in collaboration with medical experts.

10.2 Future works

We believe that the pipeline that we proposed, based on group importance scores and statistical permutation schemes, provides an interesting automatic approach to analyse neuroimaging data, as validated by our experiments in the thesis. There nevertheless remain several potential directions of improvements of this pipeline and of tree ensemble methods in the context of neuroimaging data.

We believe that additional experiments should be conducted in order to confirm the promising results obtained by the methods in Chapter 5. In particular embedded methods did not perform well and we should analyse the reason for this failure with more experiments. Several alternatives to these methods could also be explored. For example, one of the drawbacks of the Group Random Forests sampling scheme is that only one feature is drawn from each selected group, which makes the hypothesis that all features within a group are more or less equivalent. On the contrary, one could instead incorporate several or all features from each of K randomly selected groups into the evaluation. Extensions of the method that cumulates the impurity of all tested features could also be considered, exploiting more the spatial or group structure between the features (for example, by only cumulating the impurity decreases for the features that are spatially close or in the same group as the optimal one).

In this thesis, we focused exclusively on Random Forests methods. As discussed in Chapter 2, another family of tree ensemble methods is boosting which performs very well in many applications. It would be worth exploring these methods in the context of neuroimaging data and comparing them with Random Forests. In particular, it would be interesting to study adaptations of boosting methods to groups of features or spatial structure in the data. Given the link that exists between boosting and Lasso (cf. forward stagewise regression in Chapter 16 of Hastie et al. [2009]), such adaptations could be inspired from the Group Lasso method. One drawback of boosting methods however is that variable importance scores derived from such models are less well understood than importance scores derived from Random Forests. Another family of methods that could be interesting to evaluate on our datasets is deep learning. Deep learning methods have

proven their efficiency in many applications like computer vision or natural language processing. Although they are not already so familiar in the neuroimaging field because of some difficulties to interpret the methods and also due to the low sample size of the datasets encountered in this field, some research works have shown promising results for neuroimaging applications [Suk et al., 2014, Plis et al., 2014].

While most of our results are obtained with a pre-defined atlas of brain regions, we conducted some preliminary experiments with data-driven atlases in Chapter 7 (reported in Appendix B), i.e. atlases derived automatically from the data. So far, our results with such atlases were not very conclusive but we believe that it is worth pursuing this line of work. One idea here could be to adopt a hybrid approach, starting from an existing atlas and refining it, e.g. by splitting existing regions, using the data. While we only explored data-driven atlases obtained prior to the forests construction, it might make sense also to try to interleave both forest construction with the atlas derivation.

From the point of view of the applications, while we focus on Alzheimer's disease, the proposed approaches are generic and it would be surely interesting to apply them to study and improve our understanding of some other diseases. While we focus on PET images, analysing groups of features is relevant also for other modalities in neuroimaging such as fMRI data. More generally, the proposed approaches could be also tested in other application domains where features are also naturally organized spatially or in groups. One such domain is genome-wide association studies where features correspond to mutations linearly organized along the genome and a group structure is induced by haplotype blocks [Botta, 2013].

The real datasets that we exploited in this thesis are all of very small sample sizes, which makes these problems very challenging for machine learning methods. We addressed this issue in the thesis by exploiting knowledge about the structure that exists between the features in these datasets. Another interesting approach to improve performance could be to actually analyse these datasets in combination with other datasets, for example datasets made public through initiatives such as ADNI for Alzheimer's disease or the Human Brain Project for more general topics. Even if such datasets are not directly related to the question targeted by the initial small-scale dataset, we believe that they could be nevertheless exploited for example to find general correlation patterns between voxels that could help reducing data dimensionality. Such transfer learning approaches have indeed proven to be very successful in other domains such as computer vision.

Part V

Appendices

Appendix A

Coupon's collector problem

A.1 Variance of d_i

Let X a random variable following a geometric distribution of parameter p. In particular, X denotes the number of trials necessary to obtain the first success. We can write this

$$X \sim Geom(p)$$

with

$$P(X = k) = (1-p)^{k-1}p$$

is the probability that the $k-th$ trial corresponds to the first success.

The expected value of such random variable is given by $E[X] = \frac{1}{p}$. Thus, the variance of X is obtained by the following trick :

$$\begin{aligned} Var[X] &= E[X^2] - (E[X])^2 = E[X^2] - \frac{1}{p^2} && \text{(A.1)} \\ &= E[X(X-1)] + E[X] - \frac{1}{p^2}. && \text{(A.2)} \end{aligned}$$

In this equation, the expected value $E[X(X-1)]$ can be easily obtained thanks to the probability generating function of the variable X. Indeed, the probability generating function of X G(z) is given by

$$\begin{aligned} E[z^X] &= \sum_{i=1}^{\infty}(1-p)^{i-1}\ p\ z^i = \sum_{i=0}^{\infty}(1-p)^i\ p\ z^i\ z && \text{(A.3)} \\ &= zp\sum_{i=0}^{\infty}((1-p)z)^i = zp\frac{1}{1-z(1-p)} && \text{(A.4)} \end{aligned}$$

and the probability generating function is linked to the factorial moment $E[X(X-1)\dots(X-k+1)]$ such that

$$E\left[\frac{X!}{(X-k)!}\right] = G^{(k)}(1^-),\ k \geq 0.$$

From that, it follows

$$
\begin{aligned}
E[X(X-1)] &= G^{(2)}(1^-) = \frac{d^2}{dz^2}\left(\frac{zp}{1-z(1-p)}\right)\Big]_{1^-} && \text{(A.5)}\\
&= \frac{d}{dz}\left(\frac{1-z(1-p)+z(1-p)}{(1-z(1-p))^2}\right)\Big]_{1^-} && \text{(A.6)}\\
&= \frac{d}{dz}\left(\frac{p}{(1-z(1-p))^2}\right)\Big]_{1^-} && \text{(A.7)}\\
&= \left(\frac{2p(1-p)}{(1-z(1-p))^3}\right)\Big]_{1^-} && \text{(A.8)}\\
&= \frac{2(1-p)}{p^2}. && \text{(A.9)}
\end{aligned}
$$

Injecting (A.9) in (A.2) leads to

$$
\begin{aligned}
Var[X] &= \frac{2(1-p)}{p^2} + \frac{1}{p} - \frac{1}{p^2} && \text{(A.10)}\\
&= \frac{2-2p+p-1}{p^2} = \frac{1-p}{p^2}. && \text{(A.11)}
\end{aligned}
$$

A.2 Variance of D

Proposition 5. *Let K the number of features drawn without replacement at each trial and m the total number of distinct features. If K = 1, the variance of D corresponds to the following sum*

$$Var(D) = \sum_{i=1}^{m} \frac{1-p_i}{p_i^2},$$

where $p_i = \frac{m-(i-1)}{m}$, and is bounded by $\frac{\pi^2}{6}m^2$.

Proof. Indeed, we know that the random variables d_i follow a geometric distribution of parameter p_i. For such random variable, the variance $Var(d_i)$ is given by $\frac{1-p_i}{p_i^2}$ [1].

As D is the sum of the independent random variables d_i, we easily find

$$
\begin{aligned}
& Var(D) = \sum_{i=1}^{m} Var(d_i) = \sum_{i=1}^{m} \frac{1-p_i}{p_i^2} && \text{(A.12)}\\
= \; & (1-p_1)\frac{m^2}{m^2} + (1-p_2)\frac{m^2}{(m-1)^2} + (1-p_3)\frac{m^2}{(m-2)^2} + \ldots + (1-p_m)\frac{m^2}{1^2} && \text{(A.13)}\\
< \; & \frac{m^2}{m^2} + \frac{m^2}{(m-1)^2} + \frac{m^2}{(m-2)^2} + \ldots + \frac{m^2}{1^2} && \text{(A.14)}\\
< \; & m^2\frac{\pi^2}{6}. && \text{(A.15)}
\end{aligned}
$$

The line (A.13) is obtained by replacing the denominator p_i^2 in the components of the sum while line (A.14) can be claimed because $1-p_i < 1$ (for all $i = 1, \ldots, m$). Finally, the Euler's approach of the Basal problem affirms that $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$ and this affirmation is exploited from equation (A.14) to (A.15).

□

[1] The derivation of the variance of d_i is less straightforward than the one of its expected value. To not surcharge the text, we let this development at the appreciation of interested readers in the Appendix A.

A.3 K > 1 proposition

Let $K > 1$ the number of features drawn at each trial with replacement and M the total number of distinct features. The expected number of groups of K features that we need to draw in order to have seen the whole set of features is given by the following sum

$$C_M^K \sum_{i=1}^{M-K} (-1)^{i-1} \frac{C_M^i}{C_M^K - C_{M-i}^K} + \sum_{i=M-K+1}^{M} (-1)^{i-1} C_M^i.$$

We provide here a proof of this formula directly inspired from Sardy and Velenik [2010] and Ferrante and Frigo [2012].

Proof. Let N be the number of groups that should be drawn in order to obtain the whole set of features. The groups keep the same size K all over the experiment and each group has the same likelihood to be drawn. Each feature cannot appear more than once in a group. By consequence, they are C_M^K distinct groups which could be drawn and the probability of drawing any group g is given by

$$p_g = \frac{1}{C_M^K}.$$

We denote by G_i the number of groups drawn to obtain the first group with the feature i in it, for i from 1 to M. The total number of groups is thus given by $N = \max(G_1, G_2, ..., G_M)$.

These random variables G_i follow a geometric distribution. The parameter of this distribution is the probability to have a success at one trial, i.e. the probability to draw a group with the feature i. C_M^K is the total number of different groups of size K and C_{M-1}^K is the total number of groups of size K not containing the feature i. The G_i are thus following a geometric distribution of parameter

$$1 - \frac{C_{M-1}^K}{C_M^K}.$$

Therefore, the random variables $\min(G_i, G_j)$ follow a geometric distribution with parameter $1 - \frac{C_{M-2}^K}{C_M^K}$ because C_{M-2}^K represents the total number of groups not containing features i and j. In a same way, the parameter for the minimum of three random variables will be $1 - \frac{C_{M-3}^K}{C_M^K}$ and so on. For $\min(G_{i_1}, G_{i_2}, ..., G_{i_{M-K}})$, the geometric distribution parameter is $1 - \frac{1}{C_M^K}$. For $k > M - K$, it is straightforward that $\min(G_{i_1}, G_{i_2}, ..., G_{i_k})$ is a constant unitary random variable.

The cumulative distribution function of these random variables is given by

$$P(\min(G_{i_1}, G_{i_2}, ..., G_{i_k}) \leq l) = 1 - \left(\frac{C_{M-k}^K}{C_M^K} \right)^l, for\ k = (1, \ldots, M-K),$$

as they follow a geometric distribution, and thus

$$P(\min(G_{i_1}, G_{i_2}, ..., G_{i_k}) > l) = \left(\frac{C_{M-k}^K}{C_M^K} \right)^l, for\ k = (1, \ldots, M-K).$$

For $k > M - K$,

$$P(\min(G_{i_1}, G_{i_2}, ..., G_{i_k}) > 0) = 1$$

and

$$P(\min(G_{i_1}, G_{i_2}, ..., G_{i_k}) > l) = 0, for\ l \geq 1.$$

Applying the Maximum-Minimums Principle, we have

$$P(N > l) = \sum_{k=1}^{M}(-1)^{k-1} \sum_{1 \leq i_1 < i_2 < ... < i_k \leq M} P(\min(G_{i_1}, G_{i_2}, ..., G_{i_k}) > l). \quad \text{(A.16)}$$

The expected value of N is thus

$$E[N] = \sum_{l=0}^{\infty} P(N > l) \quad \text{(A.17)}$$

$$= \sum_{l=0}^{\infty} \left(\sum_{k=1}^{M-K} (-1)^{k-1} C_M^k \left(\frac{C_{M-k}^K}{C_M^K} \right)^l \right) + \sum_{k=M-K+1}^{M} (-1)^{k-1} C_M^k \quad \text{(A.18)}$$

$$= \sum_{k=1}^{M-K} (-1)^{k-1} C_M^k \frac{1}{1 - \frac{C_{M-k}^K}{C_M^K}} + \sum_{k=M-K+1}^{M} (-1)^{k-1} C_M^k \quad \text{(A.19)}$$

$$= C_M^K \sum_{k=1}^{M-K} (-1)^{k-1} \frac{C_M^k}{C_M^K - C_{M-i}^K} + \sum_{k=M-K+1}^{M} (-1)^{k-1} C_M^k. \quad \text{(A.20)}$$

□

Appendix B

Group importance scores

B.1 Real dataset

B.1.1 Tables

Table B.1 – Atlas information about the number of features per group. μ and σ stand for the average and the standard deviation of the number of features per group in the atlas respectively.

Atlas	μ	σ	Range
AAL	1431.3	1047.6	47-4791

B.1.2 Figures

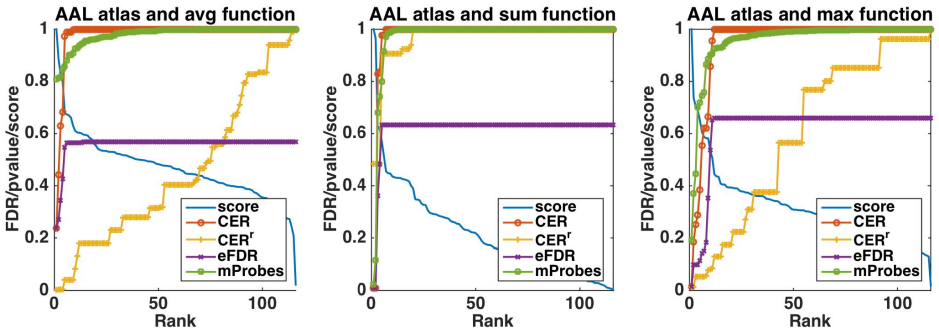


Figure B.1 – Curves of the importance scores and the different statistical scores obtained with the four methods for the AAL atlas and the CRC dataset. The curve labeled as 'Score' is the group importance score.

B.2 Data-driven atlases

To complement results in the chapter with the AAL atlases, we provide in this section results with data-driven atlases. The motivation for data-driven atlases is that pre-defined atlases, such as AAL, are in general available only for the sake of result interpretation. They are typically used to label, in terms of structurally or functionally defined brain areas, the localization of the selected voxels but they are not necessarily representative of the group structure encoded by the data itself.

We consider here two clustering techniques to derive data-driven atlases: the hierarchical agglomerative clustering approach proposed in [Thirion et al., 2014] and an original hierarchical divisive approach inspired by regression tree construction methods. The idea of this technique is to learn a regression tree to predict the signal at each voxel of PET images from its 3D coordinates. The learning sample for growing this tree is thus composed of all $n \times m$ voxels measured in the learning sample described each by three input features corresponding to their x, y, and z coordinates and one numerical output corresponding to the signal at this voxel in the PET image. The leaves of the resulting regression tree will then define the disjoint groups of voxels of the atlas. The number of brain regions is set to a user-defined value k by limiting the maximum number of splits in the tree to $k-1$ and by growing the tree using a best-first strategy (i.e., splitting at each step the leaf of highest output variance). Note that, since each tree split compares one coordinate with a threshold, the resulting groups will necessarily correspond to spatially connected brain regions as expected. A similar algorithm was exploited in [Geurts, 2001] to construct piecewise constant approximations of time series.

Both algorithms are unsupervised methods. They use information from the input matrix X to compose groups but have no concern for the labels Y.

Using these two algorithms, four atlases are derived:

- two atlases (denoted HC) obtained with hierarchical agglomerative clustering [Thirion et al., 2014], either with 116 regions, as in the AAL atlas, or with 1000 areas to test a finer resolution;
- two atlases (116 and 1000 regions) obtained with the divisive clustering approach described above, also with 116 and 1000 regions. We call this type of atlas “CART clustering".

Information about mean and standard deviation of the number of features per group for the four atlases is available in Table B.2.

Similar experiments as with AAL atlas were reproduced with these four atlases. Table B.3 shows the number of regions selected by each method and four Random forests parameter settings. These numbers follow similar trends as with the AAL atlas (in Table 7.2 of the chapter). Only very few regions are selected by all statistical scores, except CER^r which most probably suffer from a high false positive rate. Increasing the number of regions from 116 to 1000 does not necessarily increase the number of significant regions.

Analysing the groups selected with the data-driven atlases is more difficult as the corresponding brain regions have not been labelled. We attempted such analysis by looking at the AAL regions that overlap the groups ranked at the top for the data-driven atlases. The lists of these regions are reported in Tables B.4, B.5, B.6 and B.7 in the supplementary material respectively for the four data-driven atlases, CART_{116}, HC_{116}, CART_{1000}, and HC_{1000}. Lists are provided for each combination of aggregation function and Random Forests parameters. The number of top groups from the data-driven atlas

that are projected on the AAL regions was determined for each aggregation function using the maximum between the number of groups selected by CER and mProbes for this atlas with $K = \sqrt{m}$ and $T = 10,000$, as reported in the last part of Table B.3. Using the same number of groups for all Random Forests parameter combinations allows to analyse the top ranking for the data-driven atlases even when no group is actually selected by CER and mProbes for this particular combination (for example, this is the case when $K = 1$ and $T = 1000$ with all atlases). From this information, we can thus assess whether the group selection methods were right not to select any group. To simplify the discussion, let us focus the analysis on the *average* aggregation (in the top parts of the tables). The interpretation of the results obtained with the *sum* and *max* aggregations is more difficult as these two functions often lead to no group selected or the selected groups correspond to too many groups from the AAL atlases to be analysed.

For the smaller atlases, CART_{116} and HC_{116}, the selected groups with $K = \sqrt{m}$, 4 for CART_{116} and 1 for HC_{116}, overlap with 21 groups from AAL for CART_{116} and 3 groups from AAL for HC_{116}. These groups do not depend on T and they contain several regions already highlighted earlier. When $K = 1$ with the same atlases, the AAL regions remains the same for HC_{116} although they are not selected any more by CER or mProbes when $T = 1000$. For CART_{116}, there are some differences in the AAL regions that are selected, although the AAL regions at the top of the ranking are very similar. Again, when $T = 1$, no regions are selected by CER and mProbes, suggesting that these methods are too conservative. With CART_{1000}, except for $(K = 1, T = 1000)$, 3 groups are selected by CER or mProbes that overlap with 5 or 6 AAL regions. These regions match very well the regions selected using the AAL atlas and are also very consistent with the literature. Two groups are selected in the case of HC_{1000} that leads to at most 4 groups from AAL that again contains regions highlighted in the literature (angular gyrus (left)) or when using the AAL atlas (parietal inferior (right)).

Overall, although the interpretation is less straightforward, results with the data-driven atlases and the average aggregation are consistent with the results obtained with the AAL atlas. The CART atlases seem also to better match the AAL atlas than the HC atlases.

B.2.1 Tables

Table B.2 – Atlas information about the number of features per group. μ and σ stand for the average and the standard deviation of the number of features per group in the atlas respectively.

Atlas	μ	σ	Range
HC_{116}	1894.2	2248.0	10-11,007
HC_{1000}	219.7	410.1	3-2712
CART_{116}	1894.2	1433.0	224-7121
CART_{1000}	219.7	188.33	18-1848

Table B.3 – Number of regions selected ($\alpha = 0.05$) for CRC dataset for each method and each atlas, depending on the aggregation function. HC and CART stand respectively for the use of the atlas obtained with hierarchical clustering and CART clustering.

$(K;T)$	Atlas	CER			CER^r			eFDR			mProbes		
		avg	$\sum$	max	*avg*	$\sum$	max	*avg*	$\sum$	max	avg	$\sum$	max
$(1;1000)$	HC_{116}	0	0	1	0	0	3	0	0	1	0	0	0
	CART_{116}	0	0	1	10	0	6	0	0	1	0	0	0
	HC_{1000}	0	0	1	9	0	4	0	0	2	0	0	0
	CART_{1000}	0	0	1	25	0	23	0	0	1	0	0	0
$(1;10,000)$	HC_{116}	1	0	0	0	0	2	1	0	0	0	0	0
	CART_{116}	2	0	4	20	0	8	2	0	4	0	0	0
	HC_{1000}	0	1	1	4	0	6	0	1	4	0	1	0
	CART_{1000}	3	0	2	29	0	32	5	0	6	0	0	0
$(\sqrt{m};1000)$	HC_{116}	1	0	0	0	0	3	1	0	0	1	0	0
	CART_{116}	4	3	1	17	6	10	4	5	6	3	1	1
	HC_{1000}	1	0	0	44	9	29	6	0	0	0	0	0
	CART_{1000}	3	2	4	92	17	39	16	7	10	3	2	2
$(\sqrt{m};10,000)$	HC_{116}	1	0	0	>1	0	>1	1	0	0	1	0	4
	CART_{116}	4	2	2	>5	>5	>5	4	5	5	2	2	1
	HC_{1000}	2	0	0	>9	6	>9	9	0	0	1	0	2
	CART_{1000}	3	2	4	>16	>16	>16	16	10	15	3	1	3

Table B.4 – CRC dataset. First top-ranked regions of the AAL atlas corresponding to the top-ranked regions of the CART_{116} atlas selected with CER, $K = \sqrt{m}$ and $T = 10,000$, i.e. 4 region for the *avg*, 2 region for the *sum* and 2 regions for the *max*. Ranked are provided by Random Forest with different aggregation functions depending on parameters K and T. R and L stand for right and left hemisphere respectively.

	Rank	$(K;T) = (1;1,000)$	$(K;T) = (1;10,000)$	$(K;T) = (\sqrt{m};1,000)$	$(K;T) = (\sqrt{m};10,000)$
avg	1	Cerebelum Crus1 (R)	Cerebelum Crus1 (R)	Cerebelum Crus1 (R)	Cerebelum Crus1 (R)
	2	Inf. temporal g. (R)	Inf. temporal g. (R)	Inf. temporal g. (R)	Inf. temporal g. (R)
	3	Inf. occipital g. (R)	Inf. occipital g. (R)	Inf. occipital g. (R)	Inf. occipital g. (R)
	4	Mid. temporal g. (R)	Mid. temporal g. (R)	Mid. temporal g. (R)	Mid. temporal g. (R)
	5	Sup. temporal g. (R)	Sup. temporal g. (R)	Sup. temporal g. (R)	Sup. temporal g. (R)
	6	Angular g. (R)	Angular g. (R)	Angular g. (R)	Angular g. (R)
	7	Mid. occipital g. (R)	Mid. occipital g. (R)	Mid. occipital g. (R)	Mid. occipital g. (R)
	8	Parietal Inf (R)	Parietal Inf (R)	Parietal Inf (R)	Parietal Inf (R)
	9	Inf. temporal g. (L)	Fusiform (R)	SupraMarginal (R)	SupraMarginal (R)
	10	Mid. temporal g. (L)	Inf. temporal g. (L)	Postcentral (R)	Postcentral (R)
	11	Sup. temporal g. (L)	Mid. temporal g. (L)	Parietal Sup (R)	Parietal Sup (R)
	12	Rolandic Oper (L)	Sup. temporal g. (L)	Fusiform (R)	Fusiform (R)
	13	Heschl (L)	Rolandic Oper (L)	Inf. temporal g. (L)	Inf. temporal g. (L)
	14	Postcentral (L)	Heschl (L)	Mid. temporal g. (L)	Mid. temporal g. (L)
	15	SupraMarginal (L)	Postcentral (L)	Sup. temporal g. (L)	Sup. temporal g. (L)
	16	+ 10 others	+ 6 others	+ 6 others	+ 6 others
sum	1	Frontal Inf Orb (R)	Frontal Inf Orb (R)	Temporal Mid (R)	Temporal Mid (R)
	2	Frontal Mid Orb (R)	Frontal Mid Orb (R)	Temporal Inf (R)	Temporal Inf (R)
	3	Frontal Sup Orb (R)	Frontal Sup Orb (R)	Temporal Sup (R)	Temporal Sup (R)
	4	Rectus (R)	Rectus (R)	Angular g. (R)	Angular g. (R)
	5	Rectus (L)	Rectus (L)	SupraMarginal (R)	SupraMarginal (R)
	6	Frontal Sup Orb (L)	Frontal Sup Orb (L)	Parietal Inf (R)	Parietal Inf (R)
	7	Frontal Mid Orb (L)	Frontal Mid Orb (L)	Postcentral (R)	Postcentral (R)
	8	Frontal Inf Orb (L)	Frontal Inf Orb (L)	Parietal Sup (R)	Parietal Sup (R)
	9	Frontal Mid Orb (R)	Frontal Mid Orb (R)	Temporal Inf (L)	Temporal Inf (L)
	10	Frontal Mid Orb (L)	Frontal Mid Orb (L)	Temporal Mid (L)	Temporal Mid (L)
	11	Cingulum Ant (L)	Cingulum Ant (L)	Temporal Sup (L)	Temporal Sup (L)
	12	Cingulum Ant (R)	Cingulum Ant (R)	Rolandic Oper (L)	Rolandic Oper (L)
	13	Frontal Mid (R)	Frontal Mid (R)	Heschl (L)	Heschl (L)
	14	Frontal Sup Medial (L)	Frontal Sup Medial (L)	Postcentral (L)	Postcentral (L)
	15	Frontal Sup Medial (R)	Frontal Sup Medial (R)	SupraMarginal (L)	SupraMarginal (L)
	16	+ 19 others	+ 21 others	+ 2 others	+ 2 others
max	1	Temporal Mid (R)	Temporal Inf (L)	Temporal Mid (R)	Temporal Mid (R)
	2	Temporal Inf (R)	Temporal Mid (L)	Temporal Inf (R)	Temporal Inf (R)
	3	Temporal Sup (R)	Temporal Sup (L)	Temporal Sup (R)	Temporal Sup (R)
	4	Angular g. (R)	Rolandic Oper (L)	Angular g. (R)	Angular g. (R)
	5	SupraMarginal (R)	Heschl (L)	SupraMarginal (R)	SupraMarginal (R)
	6	Parietal Inf (R)	Postcentral (L)	Parietal Inf (R)	Parietal Inf (R)
	7	Postcentral (R)	SupraMarginal (L)	Postcentral (R)	Postcentral (R)
	8	Parietal Sup (R)	Angular g. (L)	Parietal Sup (R)	Parietal Sup (R)
	9	Cerebelum Crus1 (L)	Precentral (L)	Temporal Inf (L)	Cerebelum Crus1 (R)
	10	Cerebelum Crus1 (R)	Temporal Mid (R)	Temporal Mid (L)	Cerebelum 6 (R)
	11	Lingual (L)	Temporal Inf (R)	Temporal Sup (L)	Fusiform (R)
	12	Lingual (R)	Temporal Sup (R)	Rolandic Oper (L)	Occipital Inf (R)
	13	Calcarine (L)	Angular g. (R)	Heschl (L)	Occipital Mid (R)
	14	Occipital Inf (R)	SupraMarginal (R)	Postcentral (L)	Calcarine (R)
	15	Occipital Inf (L)	Parietal Inf (R)	SupraMarginal (L)	Occipital Sup (R)
	16	+ 7 others	+ 2 others	+ 2 others	+ 0 others

Table B.5 – CRC dataset. First top-ranked regions of the AAL atlas corresponding to the top-ranked regions of the HC_{116} atlas selected with mProbes, $K = \sqrt{m}$ and $T = 10,000$, i.e. 1 region for the *avg*, 0 region for the *sum* and 4 regions for the *max*. Ranked are provided by Random Forest with different aggregation functions depending on parameters K and T. R and L stand for right and left hemisphere respectively.

	Rank	$(K;T) = (1;1,000)$	$(K;T) = (1;10,000)$	$(K;T) = (\sqrt{m};1,000)$	$(K;T) = (\sqrt{m};10,000)$
avg	1	Mid. occipital g. (L)	Mid. occipital g. (L)	Mid. occipital g. (L)	Mid. occipital g. (L)
	2	Angular g. (L)	Angular g. (L)	Angular g. (L)	Angular g. (L)
	3	Angular g. (R)	Angular g. (R)	Angular g. (R)	Angular g. (R)
$\sum$					
max	1	Frontal Sup Orb (R)	Cerebelum Crus1 (L)	Frontal Inf Orb (L)	Frontal Inf Orb (L)
	2	Fusiform (L)	Cerebelum Crus1 (R)	Frontal Inf Orb (R)	Frontal Inf Orb (R)
	3	Lingual (L)	Cerebelum Crus2 (L)	Frontal Mid Orb (R)	Frontal Mid Orb (L)
	4	Lingual (R)	Cerebelum 6 (R)	Mid. temporal g. (L)	Frontal Mid Orb (R)
	5	Cerebelum 6 (R)	Vermis 7	Amygdala (L)	Sup. temporal g. (R)
	6	Vermis 6	Vermis 6	Amygdala (R)	Insula (L)
	7	Cerebelum 6 (L)	Cerebelum 6 (L)	Sup. temporal g. (L)	Sup. temporal g. (L)
	8	Rectus (R)	Inf. temporal g. (R)	Insula (L)	Temporal Pole Sup (L)
	9	Rectus (L)	Inf. temporal g. (L)	Olfactory (R)	Temporal Pole Sup (R)
	10	Frontal Sup Orb (L)	Fusiform (R)	Olfactory (L)	Mid. temporal g. (L)
	11	Frontal Inf Orb (R)	Fusiform (L)	Temporal Pole Sup (L)	Insula (R)
	12	Frontal Inf Orb (L)	Cerebelum 3 (R)	Temporal Pole Sup (R)	Hippocampus (L)
	13	Frontal Mid Orb (L)	Vermis 3	Hippocampus (R)	Caudate (R)
	14	Frontal Mid Orb (R)	Cerebelum 3 (L)	Hippocampus (L)	Caudate (L)
	15	Olfactory (R)	Vermis 1 2	Sup. temporal g. (R)	Olfactory (R)
	16	+ 86 others	+ 80 others	+ 80 others	+ 68 others

Table B.6 – CRC dataset. First top-ranked regions of the AAL atlas corresponding to the top-ranked regions of the $CART_{1000}$ atlas selected with CER, $K = \sqrt{m}$ and $T = 10,000$, i.e. 3 regions for the *avg*, 2 regions for the *sum* and 4 regions for the *max*. Ranked are provided by Random Forest with different aggregation functions depending on parameters K and T. R and L stand for right and left hemisphere respectively.

	Rank	$(K;T) = (1;1,000)$	$(K;T) = (1;10,000)$	$(K;T) = (\sqrt{m};1,000)$	$(K;T) = (\sqrt{m};10,000)$
avg	1	Inf. temporal g. (L)	Mid. temporal g. (R)	Mid. temporal g. (R)	Mid. temporal g. (R)
	2	Mid. temporal g. (L)	Sup. temporal g. (R)	Sup. temporal g. (R)	Sup. temporal g. (R)
	3	Insula (R)	Angular g. (R)	Angular g. (R)	Angular g. (R)
	4	Frontal Inf Tri (R)	SupraMarginal (R)	SupraMarginal (R)	SupraMarginal (R)
	5	Frontal Mid Orb (L)	Inf. temporal g. (R)	Parietal Inf (R)	Parietal Inf (R)
	6	Frontal Inf Orb (L)	Parietal Inf (R)		
$\sum$	1	Calcarine (R)	Calcarine (R)	Mid. temporal g. (R)	Mid. temporal g. (R)
	2	Lingual (R)	Lingual (R)	Sup. temporal g. (R)	Sup. temporal g. (R)
	3	Lingual (L)	Lingual (L)	Angular g. (R)	Angular g. (R)
	4	Calcarine (L)	Calcarine (L)	SupraMarginal (R)	SupraMarginal (R)
	5	Precuneus (L)	Precuneus (L)		
	6	Cuneus (R)	Cuneus (R)		
	7	Cuneus (L)	Cuneus (L)		
	8	Precuneus (R)	Precuneus (R)		
	9	Cingulum Mid (R)	Cingulum Mid (R)		
	10	Cingulum Mid (L)	Cingulum Mid (L)		
	11	Frontal Sup Medial (L)	Frontal Sup Medial (L)		
	12	Frontal Sup Medial (R)	Frontal Sup Medial (R)		
	13	Supp Motor Area (L)	Supp Motor Area (L)		
	14	Supp Motor Area (R)	Supp Motor Area (R)		
	15	Frontal Sup (R)	Frontal Sup (R)		
max	1	Mid. temporal g. (R)	Inf. temporal g. (L)	Mid. temporal g. (R)	Mid. temporal g. (R)
	2	Sup. temporal g. (R)	Mid. temporal g. (L)	Sup. temporal g. (R)	Sup. temporal g. (R)
	3	SupraMarginal (R)	Sup. temporal g. (L)	SupraMarginal (R)	SupraMarginal (R)
	4	Lingual (R)	Heschl (L)	Angular g. (R)	Angular g. (R)
	5	Calcarine (L)	Rolandic Oper (L)	Inf. temporal g. (L)	Inf. temporal g. (R)
	6	Lingual (L)	Postcentral (L)	Mid. temporal g. (L)	Mid. occipital g. (R)
	7	Inf. occipital g. (L)	SupraMarginal (L)	Sup. temporal g. (L)	Parietal Inf (R)
	8	+ 13 others	+ 8 others	+ 6 others	+ 0 others

Table B.7 – CRC dataset. First top-ranked regions of the AAL atlas corresponding to the top-ranked regions of the HC_{1000} atlas selected with $K = \sqrt{m}$ and $T = 10,000$, i.e. 2 regions for the *avg* with CER, 0 region for the *sum* and 2 regions for the *max* with mProbes. Ranked are provided by Random Forest with different aggregation functions depending on parameters K and T. R and L stand for right and left hemisphere respectively.

	Rank	$(K;T) = (1;1,000)$	$(K;T) = (1;10,000)$	$(K;T) = (\sqrt{m};1,000)$	$(K;T) = (\sqrt{m};10,000)$
avg	1	Frontal Inf Orb (L)	Parietal Inf (L)	Angular g. (L)	Angular g. (L)
	2	Parietal Inf (L)	Parietal Inf (R)	Angular g. (R)	Angular g. (R)
	3		Angular g. (L)		Parietal Inf (R)
	4		Angular g. (R)		
sum					
max	1	Frontal Inf Tri (R)	Inf. occipital g. (L)	Frontal Mid Orb (R)	Frontal Mid Orb (L)
	2	Pallidum (L)	Inf. occipital g. (R)	Frontal Inf Orb (L)	Frontal Inf Orb (L)
	3	Pallidum (R)	Calcarine (L)	Frontal Inf Orb (R)	Frontal Mid Orb (R)
	4	Thalamus (L)	Lingual (R)	Insula (R)	Frontal Inf Orb (R)
	5	Thalamus (R)	Fusiform (L)	Olfactory (R)	Sup. temporal g. (L)
	6	Vermis 4 5	Frontal Inf Orb (R)	Olfactory (L)	Temporal Pole Sup (L)
	7	Mid. temporal g. (R)	Frontal Inf Orb (L)	Caudate (L)	Temporal Pole Sup (R)
	8	Calcarine (L)	Frontal Mid Orb (R)	Cingulum Ant (L)	Insula (L)
	9	Mid. temporal g. (L)	Frontal Sup Orb (L)	Frontal Mid Orb (R)	Insula (R)
	10	Calcarine (R)	Frontal Mid Orb (L)	Cingulum Ant (R)	Caudate (R)
	11	Lingual (R)	Caudate (R)	Frontal Mid Orb (L)	Caudate (L)
	12	Occipital Sup (R)	Temporal Pole Sup (R)	Frontal Mid Orb (L)	Olfactory (R)
	13	Cuneus (R)	Olfactory (L)	Caudate (R)	Olfactory (L)
	14	Mid. occipital g. (R)	Caudate (L)	Putamen (R)	Lingual (L)
	15	Cingulum Ant (R)	Insula (R)	Mid. temporal g. (L)	ParaHippocampal (L)
	16	+ 26 others	+ 47 others	+ 46 others	+ 26 others

Appendix C

AAL atlas details

Table C.1 – Lists of regions in the AAL atlas with the number of voxels included in each region. The minimum group size is 47 and the maximum group size is 4791. The average size is 1431 and the median is 1170.

	AAL name	♯ voxels
1	Precentral (L)	3263
2	Precentral (R)	2651
3	Frontal Sup (L)	3471
4	Frontal Sup (R)	3368
5	Frontal Sup Orb (L)	713
6	Frontal Sup Orb (R)	556
7	Frontal Mid (L)	4705
8	Frontal Mid (R)	3925
9	Frontal Mid Orb (L)	841
10	Frontal Mid Orb (R)	732
11	Frontal Inf Oper (L)	1020
12	Frontal Inf Oper (R)	1208
13	Frontal Inf Tri (L)	2486
14	Frontal Inf Tri (R)	1559
15	Frontal Inf Orb (L)	1704
16	Frontal Inf Orb (R)	1530
17	Rolandic Oper (L)	991
18	Rolandic Oper (R)	1243
19	Supp Motor Area (L)	2034
20	Supp Motor Area (R)	2206
21	Olfactory (L)	280
22	Olfactory (R)	260
23	Frontal Sup Medial (L)	2777
24	Frontal Sup Medial (R)	1904
25	Frontal Mid Orb (L)	543
26	Frontal Mid Orb (R)	664
27	Rectus (L)	756
28	Rectus (R)	637
29	Insula (L)	1898
30	Insula (R)	1752
31	Cingulum Ant (L)	1426
32	Cingulum Ant (R)	1286

33	Cingulum Mid (L)	1940
34	Cingulum Mid (R)	2135
35	Cingulum Post (L)	466
36	Cingulum Post (R)	323
37	Hippocampus (L)	932
38	Hippocampus (R)	951
39	ParaHippocampal (L)	985
40	ParaHippocampal (R)	1109
41	Amygdala (L)	211
42	Amygdala (R)	240
43	Calcarine (L)	2248
44	Calcarine (R)	1846
45	Cuneus (L)	1434
46	Cuneus (R)	1386
47	Lingual (L)	2148
48	Lingual (R)	2321
49	Occipital Sup (L)	1245
50	Occipital Sup (R)	1343
51	Occipital Mid (L)	3186
52	Occipital Mid (R)	1963
53	Occipital Inf (L)	941
54	Occipital Inf (R)	982
55	Fusiform (L)	2227
56	Fusiform (R)	2442
57	Postcentral (L)	3614
58	Postcentral (R)	2909
59	Parietal Sup (L)	1810
60	Parietal Sup (R)	1471
61	Parietal Inf (L)	2385
62	Parietal Inf (R)	1130
63	SupraMarginal (L)	1206
64	SupraMarginal (R)	1598
65	Angular (L)	1144
66	Angular (R)	1558
67	Precuneus (L)	3222
68	Precuneus (R)	3018
69	Paracentral Lobule (L)	1069
70	Paracentral Lobule (R)	675
71	Caudate (L)	942
72	Caudate (R)	982
73	Putamen (L)	964
74	Putamen (R)	1065
75	Pallidum (L)	270
76	Pallidum (R)	256
77	Thalamus (L)	1056
78	Thalamus (R)	1025
79	Heschl (L)	224
80	Heschl (R)	222
81	Temporal Sup (L)	2265
82	Temporal Sup (R)	2583
83	Temporal Pole Sup (L)	1130
84	Temporal Pole Sup (R)	888
85	Temporal Mid (L)	4791

86	Temporal Mid (R)	3751
87	Temporal Pole Mid (L)	648
88	Temporal Pole Mid (R)	754
89	Temporal Inf (L)	2989
90	Temporal Inf (R)	2992
91	Cerebelum Crus1 (L)	2239
92	Cerebelum Crus1 (R)	2026
93	Cerebelum Crus2 (L)	1770
94	Cerebelum Crus2 (R)	1668
95	Cerebelum 3 (L)	132
96	Cerebelum 3 (R)	192
97	Cerebelum 4 5 (L)	1140
98	Cerebelum 4 5 (R)	800
99	Cerebelum 6 (L)	1714
100	Cerebelum 6 (R)	1708
101	Cerebelum 7b (L)	397
102	Cerebelum 7b (R)	316
103	Cerebelum 8 (L)	1197
104	Cerebelum 8 (R)	1204
105	Cerebelum 9 (L)	643
106	Cerebelum 9 (R)	598
107	Cerebelum 10 (L)	143
108	Cerebelum 10 (R)	127
109	Vermis 1 2	47
110	Vermis 3	230
111	Vermis 4 5	669
112	Vermis 6	368
113	Vermis 7	195
114	Vermis 8	240
115	Vermis 9	166
116	Vermis 10	105

Bibliography

I. Adler, S. Oren, and S. M. Ross. The coupon-collector's problem revisited. *Journal of Applied Probability*, 40(2):513–518, 2003.

A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.

A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 77–82. IEEE, 2007.

J. Ashburner and K. J. Friston. Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821, 2000.

J. Ashburner, K. J. Friston, et al. Nonlinear spatial normalization using basis functions. *Human brain mapping*, 7(4):254–266, 1999.

F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

N. I. Bohnen, D. S. Djang, K. Herholz, Y. Anzai, and S. Minoshima. Effectiveness and safety of 18F-FDG PET in the evaluation of dementia: a review of the recent literature. *Journal of Nuclear Medicine*, 53(1):59–71, 2012.

V. Botta. *A walk into random forests: adaptation and application to Genome-Wide Association Studies*. PhD thesis, 2013.

V. Botta, G. Louppe, P. Geurts, and L. Wehenkel. Exploiting SNP correlations within random forest for genome-wide association studies. *PloS one*, 9(4):e93379, 2014.

R. K. Brayton. On the asymptotic behavior of the number of trials necessary to complete a set with random selection. *Journal of Mathematical Analysis and Applications*, 7(1): 31–61, 1963.

L. Breiman. Bagging Predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

M. Brett, W. Penny, and S. Kiebel. Introduction to random field theory. *Human brain function*, 2, 2003.

R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. M. Arrighi. Forecasting the global burden of Alzheimer's disease. *Alzheimer's & dementia*, 3(3):186–191, 2007.

M. K. Carroll, G. A. Cecchi, I. Rish, R. Garg, and A. R. Rao. Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage*, 44(1):112–122, 2009.

R. Casanova, C. T. Whitlow, B. Wagner, J. Williamson, S. A. Shumaker, J. A. Maldjian, and M. A. Espeland. High dimensional classification of structural MRI Alzheimer's disease data based on large scale regularization. *Frontiers in neuroinformatics*, 5, 2011.

R. Casanova, F.-C. Hsu, K. M. Sink, S. R. Rapp, J. D. Williamson, S. M. Resnick, M. A. Espeland, Alzheimer's Disease Neuroimaging Initiative, et al. Alzheimer's disease risk assessment using large-scale machine learning methods. 2013.

C. Chen, A. Liaw, and L. Breiman. Using random forest to learn imbalanced data. *University of California, Berkeley*, 110, 2004.

G. Chetelat, B. Desgranges, V. De La Sayette, F. Viader, F. Eustache, and J.-C. Baron. Mild cognitive impairment Can FDG-PET predict who is to rapidly convert to Alzheimer's disease? *Neurology*, 60(8):1374–1377, 2003.

G. Chételat, F. Eustache, F. Viader, V. D. L. Sayette, A. Pélerin, F. Mézenge, D. Hannequin, B. Dupuy, J.-C. Baron, and B. Desgranges. FDG-PET measurement is more accurate than neuropsychological assessments to predict global cognitive deterioration in patients with mild cognitive impairment. *Neurocase*, 11(1):14–25, 2005.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

R. Cuingnet, E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali, O. Colliot, Alzheimer's Disease Neuroimaging Initiative, et al. Automatic classification of patients with Alzheimer's disease from structural MRI: a comparison of ten methods using the ADNI database. *neuroimage*, 56(2):766–781, 2011.

J. L. Cummings, M. Mega, K. Gray, S. Rosenberg-Thompson, D. A. Carusi, and J. Gornbein. The Neuropsychiatric Inventory comprehensive assessment of psychopathology in dementia. *Neurology*, 44(12):2308–2308, 1994.

S. De Santi, M. J. de Leon, H. Rusinek, A. Convit, C. Y. Tarshish, A. Roche, W. H. Tsui, E. Kandil, M. Boppana, K. Daisley, et al. Hippocampal formation glucose metabolism and volume losses in MCI and AD. *Neurobiology of aging*, 22(4):529–539, 2001.

R. Díaz-Uriarte and S. A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

A. V. Doumas and V. G. Papanicolaou. The coupon collector's problem revisited: asymptotics of the variance. *Advances in Applied Probability*, 44(1):166–195, 2012.

A. Drzezga, N. Lautenschlager, H. Siebner, M. Riemenschneider, F. Willoch, S. Minoshima, M. Schwaiger, and A. Kurz. Cerebral metabolic changes accompanying conversion of mild cognitive impairment into Alzheimer's disease: a PET follow-up study. *European journal of nuclear medicine and molecular imaging*, 30(8):1104–1113, 2003.

O. J. Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

P. Erdős. On a classical problem of probability theory. 1961.

M. Ferrante and N. Frigo. A note on the coupon-collector's problem with multiple arrivals and the random sampling. *arXiv preprint arXiv:1209.2667*, 2012.

M. F. Folstein, S. E. Folstein, and P. R. McHugh. "mini-mental state": a practical method for grading the cognitive state of patients for the clinician. *Journal of psychiatric research*, 12(3):189–198, 1975.

Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

G. B. Frisoni, N. C. Fox, C. R. Jack, P. Scheltens, and P. M. Thompson. The clinical use of structural MRI in Alzheimer disease. *Nature Reviews Neurology*, 6(2):67–77, 2010.

K. J. Friston, A. P. Holmes, K. J. Worsley, J.-P. Poline, C. D. Frith, and R. S. Frackowiak. Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210, 1994.

M. Ganz, D. N. Greve, B. Fischl, E. Konukoglu, A. D. N. Initiative, et al. Relevant feature set estimation with a knock-out strategy and random forests. *NeuroImage*, 122:131–148, 2015.

Y. Ge, S. Dudoit, and T. P. Speed. Resampling-based multiple testing for microarray data analysis. *Test*, 12(1):1–77, 2003.

Y. Ge, S. C. Sealfon, and T. P. Speed. Some step-down procedures controlling the false discovery rate under dependence. *Statistica Sinica*, 18(3):881, 2008.

P. Geladi and B. R. Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

P. Geurts. Pattern extraction for time series classification. In *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '01, pages 115–127, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42534-9. URL `http://dl.acm.org/citation.cfm?id=645805.670003`.

P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

M. T. Goodrich and R. Tamassia. *Data structures and algorithms in Java*. John Wiley & Sons, 2008.

K. Gosche et al. Hippocampal volume as an index of Alzheimer neuropathology findings from the nun study. *Neurology*, 58(10):1476–1482, 2002.

K. R. Gray, R. Wolz, R. A. Heckemann, P. Aljabar, A. Hammers, D. Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-region analysis of longitudinal FDG-PET for the classification of Alzheimer's disease. *Neuroimage*, 60(1):221–229, 2012.

K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, D. Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. Random forest-based similarity measures for multi-modal classification of Alzheimer's disease. *NeuroImage*, 65:167–175, 2013.

I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon and A. Elisseeff. An introduction to feature extraction. In *Feature extraction*, pages 1–25. Springer, 2006.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

K. Herholz, E. Salmon, D. Perani, J. Baron, V. Holthoff, L. Frölich, P. Schönknecht, K. Ito, R. Mielke, E. Kalbe, et al. Discrimination between Alzheimer dementia and controls by automated analysis of multicenter FDG PET. *Neuroimage*, 17(1):302–316, 2002.

J. M. Hoffman, K. A. Welsh-Bohmer, M. Hanson, B. Crain, C. Hulette, N. Earl, and R. E. Coleman. FDG PET imaging in patients with pathologically verified dementia. *Journal of Nuclear Medicine*, 41(11):1920–1928, 2000.

L. Holst. On birthday, collectors', occupancy and other classical urn problems. *International Statistical Review/Revue Internationale de Statistique*, pages 15–27, 1986.

V. A. Huynh-Thu, L. Wehenkel, and P. Geurts. Exploiting tree-based variable importances to selectively identify relevant variables. In *JMLR: Workshop and Conference proceedings*, volume 4, pages 60–73, Antwerp, 2008. Microtome Publishing.

V. a. Huynh-Thu, Y. Saeys, L. Wehenkel, and P. Geurts. Statistical interpretation of machine learning-based feature importance scores for biomarker discovery. *Bioinformatics*, 28(13):1766–1774, 2012.

C. R. Jack, D. S. Knopman, W. J. Jagust, L. M. Shaw, P. S. Aisen, M. W. Weiner, R. C. Petersen, and J. Q. Trojanowski. Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade. *The Lancet Neurology*, 9(1):119–128, 2010.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.

R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, E. Eger, F. Bach, and B. Thirion. Multiscale mining of fMRI data with hierarchical structured sparsity. *SIAM Journal on Imaging Sciences*, 5(3):835–856, 2012.

I. T. Jolliffe. Principal Component Analysis and Factor Analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

E. Kaplan. *The assessment of aphasia and related disorders*, volume 2. Lippincott Williams & Wilkins, 1983.

E. Kaplan, H. Goodglass, and S. Weintraub. *Boston naming test*. Pro-ed, 2001.

M. Karimpoor, N. Churchill, F. Tam, C. E. Fischer, T. A. Schweizer, and S. Graham. Tablet-Based Functional MRI of the Trail Making Test: Effect of Tablet Interaction Mode. *Frontiers in human neuroscience*, 11:496, 2017.

B. J. Kelley and R. C. Petersen. Alzheimer's disease and mild cognitive impairment. *Neurologic clinics*, 25(3):577–609, 2007.

S. Klöppel, C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack, J. Ashburner, and R. S. Frackowiak. Automatic classification of MR scans in Alzheimer's disease. *Brain*, 131(3):681–689, 2008.

N. Kriegeskorte, R. Goebel, and P. Bandettini. Information-based functional brain mapping. *Proceedings of the National Academy of Sciences of the United States of America*, 103(10):3863–3868, 2006.

L. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. Linden, S. J. Johnston, et al. Random subspace ensembles for fMRI classification. *Medical Imaging, IEEE Transactions on*, 29(2):531–542, 2010.

M. B. Kursa and W. R. Rudnicki. The all relevant feature selection using random forest. *arXiv preprint arXiv:1106.5112*, 2011.

S. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu. Support vector machines for temporal classification of block design fMRI data. *NeuroImage*, 26(2):317–329, 2005.

G. Langs, B. H. Menze, D. Lashkari, and P. Golland. Detecting stable distributed patterns of brain activation using Gini contrast. *NeuroImage*, 56(2):497–507, 2011.

P. Latinne, O. Debeir, and C. Decaestecker. Limiting the number of trees in random forests. *Multiple Classifier Systems*, pages 178–187, 2001.

J. Liu, S. Ji, J. Ye, et al. SLEP: Sparse learning with efficient projections. *Arizona State University*, 6:491, 2009.

G. Louppe. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439, 2013.

K. L. Lunetta, L. B. Hayward, J. Segal, and P. Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. *BMC genetics*, 5(1):32, 2004.

B. Magnin, L. Mesrob, S. Kinkingnéhun, M. Pélégrini-Issac, O. Colliot, M. Sarazin, B. Dubois, S. Lehéricy, and H. Benali. Support vector machine-based classification of Alzheimer's disease from whole-brain anatomical MRI. *Neuroradiology*, 51(2):73–83, 2009.

D. S. Marcus et al. Open Access Series of Imaging Studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of cognitive neuroscience*, 19(9):1498–1507, 2007.

J. Mazziotta, A. Toga, A. Evans, P. Fox, J. Lancaster, K. Zilles, R. Woods, T. Paus, G. Simpson, B. Pike, et al. A probabilistic atlas and reference system for the human brain: International Consortium for Brain Mapping (ICBM). *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 356(1412):1293–1322, 2001.

G. M. McKhann, D. S. Knopman, H. Chertkow, B. T. Hyman, C. R. Jack, C. H. Kawas, W. E. Klunk, W. J. Koroshetz, J. J. Manly, R. Mayeux, et al. The diagnosis of dementia due to Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. *Alzheimer's & dementia: the journal of the Alzheimer's Association*, 7(3):263–269, 2011.

L. Meier, S. Van De Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

V. Michel, E. Eger, C. Keribin, J.-B. Poline, and B. Thirion. A supervised clustering approach for extracting predictive information from brain activation images. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 7–14. IEEE, 2010.

S. Minoshima, N. L. Foster, and D. E. Kuhl. Posterior cingulate cortex in Alzheimer's disease. 1994.

S. Minoshima, B. Giordani, S. Berent, K. A. Frey, N. L. Foster, and D. E. Kuhl. Metabolic reduction in the posterior cingulate cortex in very early Alzheimer's disease. *Annals of neurology*, 42(1):85–94, 1997.

K. Möllenhoff. *Novel methods for the detection of functional brain activity using 17O MRI*. PhD thesis, University of Liege, Liege, Belgium, 2016.

E. Moradi, A. Pepe, C. Gaser, H. Huttunen, J. Tohka, Alzheimer's Disease Neuroimaging Initiative, et al. Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects. *Neuroimage*, 104:398–412, 2015.

S. Morbelli, A. Piccardo, G. Villavecchia, B. Dessi, A. Brugnolo, A. Piccini, A. Caroli, G. Frisoni, G. Rodriguez, and F. Nobili. Mapping brain morphological and functional conversion patterns in amnestic MCI: a voxel-based MRI and FDG-PET study. *European journal of nuclear medicine and molecular imaging*, 37(1):36, 2010.

J. C. Morris. The Clinical Dementia Rating (CDR): current version and scoring rules. *Neurology*, 1993.

L. Mosconi. Brain glucose metabolism in the early and specific diagnosis of Alzheimer's disease. *European journal of nuclear medicine and molecular imaging*, 32(4):486–510, 2005.

L. Mosconi, W. H. Tsui, K. Herholz, A. Pupi, A. Drzezga, G. Lucignani, E. M. Reiman, V. Holthoff, E. Kalbe, S. Sorbi, et al. Multicenter standardized 18F-FDG PET diagnosis of mild cognitive impairment, Alzheimer's disease, and other dementias. *Journal of nuclear medicine*, 49(3):390–398, 2008.

L. Mosconi, R. Mistur, R. Switalski, W. H. Tsui, L. Glodzik, Y. Li, E. Pirraglia, S. De Santi, B. Reisberg, T. Wisniewski, et al. FDG-PET changes in brain glucose metabolism from normal cognition to pathologically verified Alzheimer's disease. *European journal of nuclear medicine and molecular imaging*, 36(5):811–822, 2009.

J. Mourão-Miranda, A. L. Bokde, C. Born, H. Hampel, and M. Stetter. Classifying brain states and determining the discriminating activation patterns: support vector machine on functional MRI data. *NeuroImage*, 28(4):980–995, 2005.

H. M. Nielsen, K. Chen, W. Lee, Y. Chen, R. J. Bauer, E. Reiman, R. Caselli, and G. Bu. Peripheral apoE isoform levels in cognitively normal APOE $\varepsilon 3/\varepsilon 4$ individuals are associated with regional gray matter volume and cerebral glucose metabolism. *Alzheimer's research & therapy*, 9(1):5, 2017.

R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér. Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, 8(Mar): 589–612, 2007.

G. Orrù, W. Pettersson-Yeo, A. F. Marquand, G. Sartori, and A. Mechelli. Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neuroscience & Biobehavioral Reviews*, 36(4):1140–1152, 2012.

T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? In *MLDM*, pages 154–168. Springer, 2012.

M. Pagani, F. De Carli, S. Morbelli, J. Öberg, A. Chincarini, G. Frisoni, S. Galluzzi, R. Perneczky, A. Drzezga, B. van Berckel, et al. Volume of interest-based [18 F] fluorodeoxyglucose PET discriminates MCI converting to Alzheimer's disease from healthy controls. A European Alzheimer's Disease Consortium (EADC) study. *NeuroImage: Clinical*, 7:34–42, 2015.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

J. Paul and P. Dupont. Inferring statistically significant features from random forests. *Neurocomputing*, 150:471–480, 2015.

W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Academic press, 2011a.

W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Academic press, 2011b.

R. C. Petersen and S. Negash. Mild cognitive impairment: an overview. *CNS spectrums*, 13(1):45–53, 2008.

R. Pfeffer, T. Kurosaki, C. Harrah Jr, J. Chance, and S. Filos. Measurement of functional activities in older adults in the community. *Journal of gerontology*, 37(3):323–329, 1982.

S. M. Plis, D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner, and V. D. Calhoun. Deep learning for neuroimaging: a validation study. *Frontiers in neuroscience*, 8, 2014.

J. L. Prince and J. M. Links. *Medical imaging signals and systems*. Pearson Prentice Hall Upper Saddle River, New Jersey, 2006.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simple MKL. *Journal of Machine Learning Research*, 9(Nov):2491–2521, 2008.

S. Rathore, M. Habes, M. A. Iftikhar, A. Shacklett, and C. Davatzikos. A review on neuroimaging-based classification studies and associated feature extraction methods for Alzheimer's disease and its prodromal stages. *NeuroImage*, 2017.

R. M. Reitan. Validity of the Trail Making Test as an indicator of organic brain damage. *Perceptual and motor skills*, 8(3):271–276, 1958.

A. Rey. L'examen psychologique dans les cas d'encéphalopathie traumatique.(les problemes.). *Archives de psychologie*, 1941.

J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville. Brain decoding of fMRI connectivity graphs using decision tree ensembles. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1137–1140. IEEE, 2010.

B. Rosén. On the coupon collector's waiting time. *The Annals of Mathematical Statistics*, pages 1952–1969, 1970.

W. G. Rosen, R. C. Mohs, and K. L. Davis. A new rating scale for Alzheimer's disease. *The American journal of psychiatry*, 1984.

S. Ryali, K. Supekar, D. A. Abrams, and V. Menon. Sparse logistic regression for whole-brain classification of fMRI data. *NeuroImage*, 51(2):752–764, 2010.

Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 313–325. Springer, 2008.

S. Sardy and Y. Velenik. Petite collection d'informations utiles pour collectionneur compulsif. *Images des Mathématiques*, pages http–images, 2010.

M. Schmidt et al. *Rey auditory verbal learning test: A handbook*. Western Psychological Services Los Angeles, CA, 1996.

J. Schrouff, J. Cremers, G. Garraux, L. Baldassarre, J. Mourão-Miranda, and C. Phillips. Localizing and comparing weight maps generated from linear kernel machine learning models. In *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*, pages 124–127. IEEE, 2013a.

J. Schrouff, M. J. Rosa, J. M. Rondina, A. F. Marquand, C. Chu, J. Ashburner, C. Phillips, J. Richiardi, and J. Mourão-Miranda. PRoNTo: pattern recognition for neuroimaging toolbox. *Neuroinformatics*, 11(3):319–337, 2013b.

J. Schrouff, J. Monteiro, L. Portugal, M. Rosa, C. Phillips, and J. Mourão-Miranda. Embedding Anatomical or Functional Knowledge in Whole-Brain Multiple Kernel Learning Models. *Neuroinformatics*, pages 1–27, 2018.

F. Segovia, C. Bastin, E. Salmon, J. M. Górriz, J. Ramírez, and C. Phillips. Combining PET images and neuropsychological test data for automatic diagnosis of Alzheimer's disease. *PloS one*, 9(2):e88687, 2014.

F. Segovia, J. G'orriz, J. Ram'ırez, C. Phillips, Alzheimer's Disease Neuroimaging Initiative, et al. Combining feature extraction methods to assist the diagnosis of Alzheimer's disease. *Current Alzheimer research*, 2015.

R. Sheldon et al. *A first course in probability*. Pearson Education India, 2002.

D. Shen, G. Wu, and H.-I. Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

D. H. Silverman, G. W. Small, C. Y. Chang, C. S. Lu, M. A. K. de Aburto, W. Chen, J. Czernin, S. I. Rapoport, P. Pietrini, G. E. Alexander, et al. Positron emission tomography in evaluation of dementia: regional brain metabolism and long-term outcome. *Jama*, 286(17):2120–2127, 2001.

W. Stadje. The collector's problem with group drawings. *Advances in Applied Probability*, 22(4):866–882, 1990.

J. D. Storey and R. Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8 (1):25, 2007.

H.-I. Suk and D. Shen. Deep learning-based feature representation for ad/mci classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 583–590. Springer, 2013.

H.-I. Suk, S.-W. Lee, D. Shen, A. D. N. Initiative, et al. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, 101:569–582, 2014.

A. Sutera, C. Châtel, G. Louppe, L. Wehenkel, and P. Geurts. Random Subspace with Trees for Feature Selection Under Memory Constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 929–937, 2018.

B. Thirion, G. Varoquaux, E. Dohmatob, and J.-B. Poline. Which fMRI clustering gives good brain parcellations? *Frontiers in neuroscience*, 8, 2014.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

T. N. Tombaugh and N. J. McIntyre. The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society*, 40(9):922–935, 1992.

E. Tuv, A. Borisov, G. Runger, and K. Torkkola. Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research*, 10(Jul):1341–1366, 2009.

N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1):273–289, 2002.

G. Varoquaux, P. R. Raamana, D. A. Engemann, A. Hoyos-Idrobo, Y. Schwartz, and B. Thirion. Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. *NeuroImage*, 2016.

P. Vemuri, J. L. Gunter, M. L. Senjem, J. L. Whitwell, K. Kantarci, D. S. Knopman, B. F. Boeve, R. C. Petersen, and C. R. Jack. Alzheimer's disease diagnosis in individual subjects using structural MR images: Validation studies. *Neuroimage*, 39(3):1186–1197, Feb 2008. doi: 10.1016/j.neuroimage.2007.09.073. URL `http://dx.doi.org/10.1016/j.neuroimage.2007.09.073`.

D. Wackerly, W. Mendenhall, and R. Scheaffer. *Mathematical statistics with applications*. Nelson Education, 2007.

D. Wechsler. *Wechsler memory scale-revised (WMS-R)*. Psychological Corporation, 1987.

D. Wechsler and M. M. De Lemos. *Wechsler adult intelligence scale-revised*. Harcourt Brace Jovanovich, 1981.

M. Wehenkel, C. Bastin, C. Phillips, and P. Geurts. Tree ensemble methods and parcelling to identify brain areas related to Alzheimer's disease. In *Pattern Recognition in Neuroimaging (PRNI), 2017 International Workshop on*, pages 1–4. IEEE, 2017.

M. Wehenkel, C. Bastin, P. Geurts, and C. Phillips. Computer Aided Diagnosis System Based on Random Forests for the Prognosis of Alzheimer's Disease. In *1st HBP Student Conference - Transdisciplinary Research Linking Neuroscience, Brain Medicine and Computer Science*, pages 14–18. Frontiers Media S.A., 2018a.

M. Wehenkel, A. Sutera, C. Bastin, P. Geurts, and C. Phillips. Random Forests based group importance scores and their statistical interpretation: application for Alzheimer's disease. *Frontiers in Neuroscience*, 12:411, 2018b. doi: 10.3389/fnins.2018.00411.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

D. Zhang, Y. Wang, L. Zhou, H. Yuan, D. Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multimodal classification of Alzheimer's disease and mild cognitive impairment. *Neuroimage*, 55(3):856–867, 2011.

D. Zhang, D. Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer's disease. *Neuroimage*, 59(2):895–907, 2012.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.